****

**API REFERANS DOKÜMANTASYONU**

n11.com API Kurumsal Servisleri, Metotları ve Kullanım Örnekleri

|  |  |
| --- | --- |
| **Change Log** | |
| 20.05.2013 | Doküman ilk versiyon oluşturuldu |
| 24.05.2013 | <status> tag içeriği değişti , data şablonlarına yeni tanımlar eklendi , ShipmentService response tarafındaki alan isimleri güncellendi. |
| 07.06.2013 | Order detail servisine itemlist>item tagı altına “sellerInvoiceAmount” alanı ve buyer tagı altında taxId,taxOffice ve email alanları eklendi. |
| 26.06.2013 | SaveProduct servisine product attributes alanı eklendi. |
| 30.07.2013 | ProductSaleStatus durumları revize edildi. |
| 17.12.2013 | OrderItemReject, OrderItemAccept, MakeOrderItemShipment servislerinde birden fazla order item için işlem yapabilmek için düzenlendi. |
| 21.05.2014 | stockItems.stockItem.optionPrice alanı eklendi. |
| 30.05.2014 | Bütün servislerin güncellemeleri yapıldı. |
| 08.08.2014 | SaveProduct servisine specialProductInfoList alanı eklendi |
| 22.08.2014 | Örnek Java Request Kodları eklendi |
| 10.03.2015 | ProductAnswer Servisi Eklendi |
| 30.09.2015 | Delivery Fee Type alanı şartlı kargo seçeneği eklendi |
| 30.09.2015 | Category Service Authentication eklendi |
| 07.11.2016 | GetCategoryAttributesId, GetCategoryAttributeValue metodları için tanımlar ve örnek istekler eklendi. |
| 17.11.2016 | Ödeme tiplerine Masterpass eklendi. |
| 26.12.2016 | SaveProduct servisine Bundle, Gtin, Mpn alanları eklendi. |
| 03.03.2017 | GetNeighborhoods metodu eklendi, DetailedOrderList ve OrderDetail alanlarına neighborhood alanı eklendi. Sipariş servislerinde searchData.status alanına “CLAIMED” eklendi. |

03.03.2017

Doğuş Planet Bilgi Teknolojileri San. Ve Tic. A.Ş

İçindekiler

[1. GİRİŞ 4](#_Toc476303375)

[2. DOĞRULAMA VE GÜVENLİK MEKANİZMASI 4](#_Toc476303376)

[2.1 Güvenlik Mekanizması 4](#_Toc476303377)

[3. METOT BİLGİLERİ 6](#_Toc476303378)

[3.1 Kategori Servisleri (CategoryService) 6](#_Toc476303379)

[Kategori Özellik ve Değer Gösterme (GetCategoryAttributes) 6](#_Toc476303380)

[Kategori Özellik Gösterme (GetCategoryAttributesId) 8](#_Toc476303381)

[Kategori Özellik Gösterme (GetCategoryAttributeValue) 10](#_Toc476303382)

[Üst Kategori Bilgisini Gösterme (GetParentCategory) 12](#_Toc476303383)

[Alt Kategori Bilgilerini Listeleme (GetSubCategories) 13](#_Toc476303384)

[Üst Kategori Bilgileri (GetTopLevelCategories) 15](#_Toc476303385)

[3.2 Şehir Servisleri (CityService) 17](#_Toc476303386)

[Şehirleri Listeleme (GetCities) 17](#_Toc476303387)

[Şehir Bilgisi Gösterme (GetCity) 18](#_Toc476303388)

[İlçe Bilgilerinin Gösterimi (GetDistrict) 20](#_Toc476303389)

[Semt/Mahalle Bilgilerinin Gösterimi (GetNeighborhoods) 21](#_Toc476303390)

[3.3 Ürün Servisleri (ProductService) 23](#_Toc476303391)

[N11 Ürün ID si ile Ürün Bilgilerini Alma (GetProductByProductId) 23](#_Toc476303392)

[Mağaza Ürün Kodu ile Ürün Bilgilerini Alma (GetProductBySellerCode) 26](#_Toc476303393)

[Ürün Listeleme (GetProductList) 30](#_Toc476303394)

[Ürün Kaydetme (Save Product) 32](#_Toc476303395)

[Ürün Arama (SearchProducts) 38](#_Toc476303396)

[N11 Ürün ID’si ile Ürün Silme (DeleteProductById) 40](#_Toc476303397)

[Mağaza Ürün Kodu ile Ürün Silme(DeleteProductBySellerCode) 42](#_Toc476303398)

[N11 Ürün ID’si ile İndirim Tutarı Güncelleme (UpdateDiscountValueByProductId) 43](#_Toc476303399)

[Mağaza Ürün Kodu ile İndirim Tutarı Güncelleme (UpdateDiscountValueBySellerCode) 46](#_Toc476303400)

[N11 Ürün Kodu ile Ürün Fiyatı Güncelleme (UpdateProductPriceById) 48](#_Toc476303401)

[Mağaza Ürün Kodu ile Ürün Fiyatı Güncelleme (UpdateProductPriceBySellerCode) 50](#_Toc476303402)

[Temel Ürün Özellikleri Güncelleme (UpdateProductBasic) 52](#_Toc476303403)

[Ürün Sorularını Listeleme (GetProductQuestionList) 55](#_Toc476303404)

[Ürün Sorularını Detaylandırma (GetProductQuestionDetail) 57](#_Toc476303405)

[Ürün Sorularını Cevaplandırma (SaveProductAnswer) 59](#_Toc476303406)

[3.4 Ürün Satış Servisleri (ProductSellingService) 60](#_Toc476303407)

[N11 Ürün ID’si ile Ürünün Satışını Başlat (StartSellingProductByProductId) 60](#_Toc476303408)

[Mağaza Ürün Kodu ile Ürünün Satışını Başlat (StartSellingProductBySellerCode) 62](#_Toc476303409)

[N11 Ürün ID’si ile Ürünün Satışını Durdur (StopSellingProductByProductId) 64](#_Toc476303410)

[Mağaza Ürün Kodu ile Ürünün Satışını Durdur (StopSellingProductBySellerCode) 65](#_Toc476303411)

[3.5 Ürün Stok Servisleri (ProductStockService) 67](#_Toc476303412)

[N11 Ürün ID si ile Stok Bilgisi Listeleme (GetProductStockByProductId) 67](#_Toc476303413)

[Mağaza Ürün Kodu ile Stok Bilgisi Listeleme (GetProductStockBySellerCode) 69](#_Toc476303414)

[Stok Özellikleri ile Stok Miktarı Güncelleme (DeleteAndUpdateStockByStockAttributes ) 71](#_Toc476303415)

[N11 Ürün Stok ID ile Stok Güncelleme (UpdateStockByStockId) 73](#_Toc476303416)

[Mağaza Stok Kodu ile Stok Güncelleme (UpdateStockByStockSellerCode) 76](#_Toc476303417)

[Stok Özellikleri ile Stok Miktarı Artırma (IncreaseStockByStockAttributes) 78](#_Toc476303418)

[N11 Ürün Stok ID ile Stok Miktarı Artırma (IncreaseStockByStockId) 80](#_Toc476303419)

[Mağaza Stok Kodu ile Stok Miktarı Arttırma (IncreaseStockByStockSellerCode) 82](#_Toc476303420)

[3.6 Sipariş Servisleri (OrderService) 84](#_Toc476303421)

[Detaylı Sipariş Listeleme (DetailedOrderList) 84](#_Toc476303422)

[Sipariş Listeleme (OrderList) 89](#_Toc476303423)

[Sipariş Detayı (OrderDetail) 91](#_Toc476303424)

[Sipariş Maddesinin Kabul Edilmesi (OrderItemAccept) 96](#_Toc476303425)

[Sipariş Kaleminin Reddedilmesi (OrderItemReject) 99](#_Toc476303426)

[Sipariş Kaleminin Kargoya Verilmesi (MakeOrderItemShipment) 102](#_Toc476303427)

[3.7 Kargo Şirketi Servisi (ShipmentCompanyService) 105](#_Toc476303428)

[Kargo Şirketlerini Listeleme (GetShipmentCompanies) 105](#_Toc476303429)

[3.8 Teslimat Şablonu Servisi (ShipmentService) 107](#_Toc476303430)

[Teslimat Şablonu Listeleme (GetShipmentTemplate) 107](#_Toc476303431)

[Teslimat Şablonu Oluşturma, Güncelleme (CreateOrUpdateShipmentTemplate) 110](#_Toc476303432)

[Teslimat Şablonu Listeleme (GetShipmentTemplateList) 115](#_Toc476303433)

[3.9 Genel Olarak Kullanılan Data Şablonları 119](#_Toc476303434)

[Yetkilendirme (Authentication) 119](#_Toc476303435)

[Sonuç Durum Bilgisi (Result) 119](#_Toc476303436)

[Sayfalandırma (Pagination) 119](#_Toc476303437)

[Ürün Özet Bilgisi (Product Summary Data) 120](#_Toc476303438)

[Ürün Satış Durumu (Product Sale Status) 120](#_Toc476303439)

[Ürün Onay Durumu (Product Approval Status) 121](#_Toc476303440)

[Ürün İndirim Bilgisi (Product Discount Data) 121](#_Toc476303441)

[Ürün Durumu (Product Condition) 121](#_Toc476303442)

[Onay Durumu (Approval Status) 121](#_Toc476303443)

[Ürün Başlığı (Title) 121](#_Toc476303444)

[Ürün Altbaşlığı (Subtitle) 121](#_Toc476303445)

[İndirim Tipi (Discount Type) 122](#_Toc476303446)

[Sipariş Durumu (SearchData) 122](#_Toc476303447)

[Ödeme Tipi (Payment Type) 122](#_Toc476303448)

[Kargo Ödeme Yöntemi (Delivery Fee Type) 122](#_Toc476303449)

[Servis Sipariş Kalemi Tipi (Orderitem Type) 122](#_Toc476303450)

[Sipariş Durum Bilgisi (Order Status) 123](#_Toc476303451)

[Sipariş Maddesi Durum Bilgisi (OrderItem Status) 123](#_Toc476303452)

[Ödeme Tipi (Payment Type) 123](#_Toc476303453)

[Kargo Çoklu Gönderim Durumu (Shipment Allowed) 124](#_Toc476303454)

# 1. GİRİŞ

Bu dokümanda, n11 kullanıcılarının site ya da Mağaza Yönetim Merkezi üzerinden yapabildikleri işlemlerin birçoğunu kapsayan API entegrasyonu bilgileri yer almaktadır. Servisler, SOAP tabanlı web servislerdir.

Web servis tanım dosyalarının adresleri aşağıda listelenmiştir.

<https://api.n11.com/ws/CategoryService.wsdl>

<https://api.n11.com/ws/CityService.wsdl>

<https://api.n11.com/ws/ProductService.wsdl>

<https://api.n11.com/ws/ProductSellingService.wsdl>

<https://api.n11.com/ws/ProductStockService.wsdl>

<https://api.n11.com/ws/OrderService.wsdl>

<https://api.n11.com/ws/ShipmentCompanyService.wsdl>

<https://api.n11.com/ws/ShipmentService.wsdl>

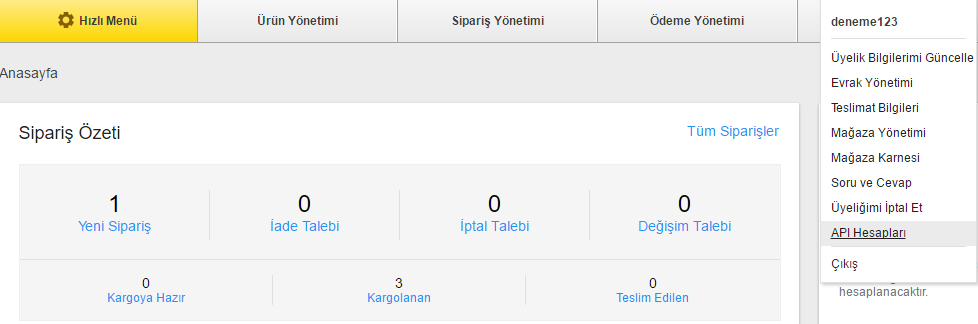
# 2. DOĞRULAMA VE GÜVENLİK MEKANİZMASI

## 2.1 Güvenlik Mekanizması

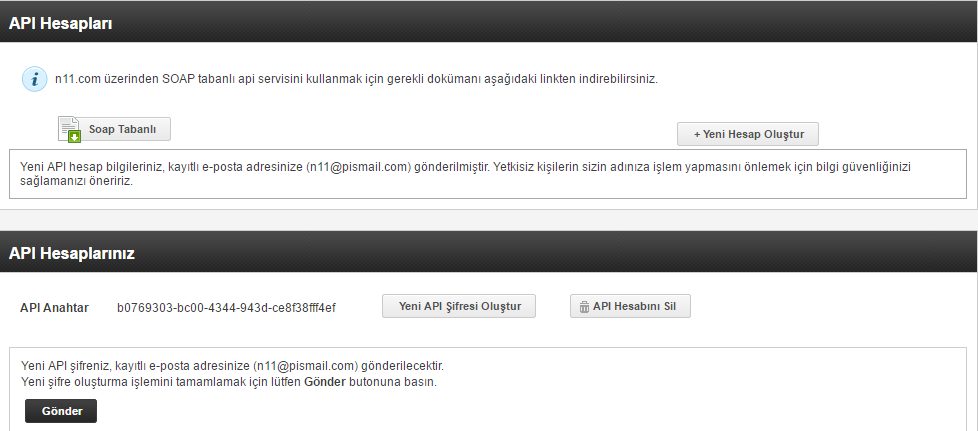
Güvenlik mekanizması olarak;

* Kimlik doğrulama: Mağaza Yönetim Merkezi’ne giriş yaptıktan sonra ana menü altındaki Hesabım – API Hesapları bölümünden alınan güvenlik anahtarının **(API Key)** her istekte gönderilmesi ile sağlanmaktadır. API şifresi, kullanıcının e-posta adresine gönderilir. API Anahtarı ve API Şifresini göndermek zorunludur. Bu iki parametre ile birlikte mağazanın kimliği tespit edilir. Bu sayede mağazanın sadece kendisine ait ürünleri ve diğer bilgileri sorgulaması sağlanır.

1) API anahtarı ve şifresi / Hesabım – API Hesapları menüsü



2) API Hesapları altından “Yeni Hesap Oluştur” fonksiyonu ile API hesabı yaratılır. Yaratılan hesabın şifresi e-posta ile gönderilir. İstenildiği takdirde API şifresi yeniden oluşturulabilir ya da API hesabı silinebilir.



3) Elde edilen API Anahtarı ve API şifresi, istenildiğinde gönderilerek yetkili kullanıcının ilgili servisi çağırdığı doğrulanır ve servis çağrıları başarılı bir şekilde sonlanır.

# 3. METOT BİLGİLERİ

## 3.1 Kategori Servisleri (CategoryService)

Kategori servisleri için WSDL adresi:

<https://api.n11.com/ws/CategoryService.wsdl>

### Kategori Özellik ve Değer Gösterme (GetCategoryAttributes)

*İstenilen kategori, üst seviye kategori veya diğer seviye kategorilerden olabilir, bu kategorilere ait olan özelliklerin ve bu özelliklere ait değerlerin listelenmesi için kullanılan metottur.*

**GetCategoryAttributesRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| **categoryId** | Özelliklerinin listelenmesi istenen kategori ID |

**\*Bold alanlar zorunlu alanlardır.**

**GetCategoryAttributesResponse**

|  |  |
| --- | --- |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| category.attributeList.attribute.id | Özelliğe sistemimizde verilen id bilgisi |
| category.attributeList.attribute.mandatory | Her bir özelliğin girilme zorunluluğu bilgisi bu alan aracılığı ile öğrenilir. |
| category.attributeList.attribute.multipleSelect | Bir özelliğin birden fazla değere sahip olup olamayacağı bilgisi bu alandan öğrenilir. |
| category.attributeList.attribute.name | Özellik adı bilgisini içerir |
| category.attributeList.attribute.valueList.value.id | Özelliğe girilecek olan değer kodunu içerir |
| category.attributeList.attribute.valueList.value.name | Özelliğe girilecek olan isim bilgisini içerir |

**GetCategoryAttributes Örnek Çağrı**

**Request**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<soapenv:Body>

<sch:GetCategoryAttributesRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<categoryId>1002306</categoryId>

</sch:GetCategoryAttributesRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetCategoryAttributesResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<category>

<attributeList>

<attribute>

<id>354189900</id>

<mandatory>true</mandatory>

<multipleSelect>false</multipleSelect>

<name>Marka</name>

<valueList>

<value>

<id>7915478</id>

<name>Axcess</name>

</value>

<value>

<id>7565492</id>

<name>Micro Genius</name>

</value>

<value>

<id>7561463</id>

<name>Microsoft</name>

</value>

<value>

<id>7561464</id>

<name>Nintendo</name>

</value>

<value>

<id>7561461</id>

<name>Sony</name>

</value>

<value>

<id>7561465</id>

<name>Square Enix</name>

</value>

<value>

<id>7918563</id>

<name>Thrustmaster</name>

</value>

</valueList>

</attribute>

</attributeList>

<id>1002306</id>

<name>Video Oyun &amp; Konsol</name>

</category>

</ns3:GetCategoryAttributesResponse>

</env:Body>

</env:Envelope>

**GetCategoryAttributes Örnek Çağrı (Java)**

public static void main(String[] args) {  
  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 long categoryIdValue = 1002806L;  
   
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 GetCategoryAttributesRequest request = new GetCategoryAttributesRequest();  
 request.setAuth(authentication);  
 request.setCategoryId(categoryIdValue);  
  
 CategoryServicePort port = new CategoryServicePortService().getCategoryServicePortSoap11();  
 GetCategoryAttributesResponse response = port.getCategoryAttributes(request);  
  
 CategoryAttributeList categoryAttributes = response.getCategory().getAttributeList();  
  
 List<CategoryAttributeData> attributeData = categoryAttributes.getAttribute();  
  
 for (CategoryAttributeData sampleAttributeData : attributeData  
 ) {  
 List<CategoryAttributeValueData> sampleAttValueDataList = sampleAttributeData.getValueList().getValue();  
 for (CategoryAttributeValueData sampleAttValueData : sampleAttValueDataList  
 ) {  
 System.out.println(sampleAttributeData.getId() + " " + sampleAttributeData.getName() + " " + sampleAttValueData.getName() + " (" + sampleAttValueData.getId() + ", " + sampleAttValueData.getDependedName() + ")");  
 }  
  
 }  
}

### Kategori Özellik Gösterme (GetCategoryAttributesId)

*İstenilen kategori, üst seviye kategori veya diğer seviye kategorilerden olabilir, bu kategorilere ait olan özelliklerin listelenmesi için kullanılan metoddur.*

**GetCategoryAttributesIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| **categoryId** | Özelliklerinin listelenmesi istenen kategori ID |

**\*Bold alanlar zorunlu alanlardır.**

**GetCategoryAttributesIdResponse**

|  |  |
| --- | --- |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| category.attributeList.attribute.id | Özelliğe sistemimizde verilen id bilgisi |
| category.attributeList.attribute.mandatory | Her bir özelliğin girilme zorunluluğu bilgisi bu alan aracılığı ile öğrenilir. |
| category.attributeList.attribute.multipleSelect | Bir özelliğin birden fazla değere sahip olup olamayacağı bilgisi bu alandan öğrenilir. |
| category.attributeList.attribute.name | Özellik adı bilgisini içerir |

**GetCategoryAttributesId Örnek Çağrı**

**Request**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<soapenv:Body>

<sch:GetCategoryAttributesIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<categoryId>1003222</categoryId>

</sch:GetCategoryAttributesIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetCategoryAttributesIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<categoryProductAttributeList>

<categoryProductAttribute>

<id>354080393</id>

<mandatory>true</mandatory>

<multipleSelect>false</multipleSelect>

<name>Marka</name>

</categoryProductAttribute>

<categoryProductAttribute>

<id>354205912</id>

<mandatory>false</mandatory>

<multipleSelect>false</multipleSelect>

<name>Gram</name>

</categoryProductAttribute>

</categoryProductAttributeList>

</ns3:GetCategoryAttributesIdResponse>

</env:Body>

</env:Envelope>

**GetCategoryAttributesId Örnek Çağrı (Java)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

long categoryIdVal = 1003222;

GetCategoryAttributesIdRequest request = new GetCategoryAttributesIdRequest();

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

request.setAuth(authentication);

request.setCategoryId(categoryIdVal);

CategoryServicePort port = new CategoryServicePortService().getCategoryServicePortSoap11();

GetCategoryAttributesIdResponse response = port.getCategoryAttributesId(request);

List<CategoryProductAttributeData> valueList = response.getCategoryProductAttributeList().getCategoryProductAttribute();

for (CategoryProductAttributeData sampleValue:valueList

) {

System.out.println("Özellik ID : " + sampleValue.getId() + " Özellik Adı: " + sampleValue.getName());

}

*}*

### Kategori Özellik Gösterme (GetCategoryAttributeValue)

Özelliğe sistemimizde verilen id bilgisini (category.attributeList.attribute.id) girdi vererek, o özelliğe ait değerleri listeler.

**GetCategoryAttributeValueRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| **categoryProductAttributeId** | Ürün özelliğine sistemimizde verilen id bilgisi |

**\*Bold alanlar zorunlu alanlardır.**

**GetCategoryAttributeValueResponse**

|  |  |
| --- | --- |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| categoryProductAttributeValueList. categoryProductAttributeValue.id | Bir özelliğe ait değer bilgisine sistemimizde verilen ID değeri |
| categoryProductAttributeValueList. categoryProductAttributeValue.name | Bir özelliğin değerine ait isim bilgisi |

**GetCategoryAttributeValue Örnek Çağrı**

**Request**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/><soapenv:Body>

<sch:GetCategoryAttributeValueRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<categoryProductAttributeId>354080997</categoryProductAttributeId>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

</pagingData>

</sch:GetCategoryAttributeValueRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetCategoryAttributeValueResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<categoryProductAttributeValueList>

<categoryProductAttributeValue>

<id>7110629</id>

<name>Tek Hoparlör</name>

</categoryProductAttributeValue>

<categoryProductAttributeValue>

<id>7110630</id>

<name>2 + 1</name>

</categoryProductAttributeValue>

<categoryProductAttributeValue>

<id>7110631</id>

<name>5 + 1</name>

</categoryProductAttributeValue>

<categoryProductAttributeValue>

<id>7110632</id>

<name>4 + 1</name>

</categoryProductAttributeValue>

<categoryProductAttributeValue>

<id>7110633</id>

<name>1 + 1</name>

</categoryProductAttributeValue>

<categoryProductAttributeValue>

<id>7110634</id>

<name>2 + 0</name>

</categoryProductAttributeValue>

<categoryProductAttributeValue>

<id>7110635</id>

<name>7 + 1</name>

</categoryProductAttributeValue>

</categoryProductAttributeValueList>

</ns3:GetCategoryAttributeValueResponse>

</env:Body>

</env:Envelope>

**GetCategoryAttributeValue Örnek Çağrı (Java)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

int currentPageVal = 0;

int pageSizeVal = 100;

long catProdAttrIdVal = 354080997;

GetCategoryAttributeValueRequest request = new GetCategoryAttributeValueRequest();

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

request.setAuth(authentication);

RequestPagingData pagingData = new RequestPagingData();

pagingData.setCurrentPage(currentPageVal);

pagingData.setPageSize(pageSizeVal);

request.setPagingData(pagingData);

request.setCategoryProductAttributeId(catProdAttrIdVal);

CategoryServicePort port = new CategoryServicePortService().getCategoryServicePortSoap11();

GetCategoryAttributeValueResponse response = port.getCategoryAttributeValue(request);

List<CategoryProductAttributeValueData> valueList = response.getCategoryProductAttributeValueList().getCategoryProductAttributeValue();

for (CategoryProductAttributeValueData sampleValue:valueList

) {

System.out.println("Değer ID : " + sampleValue.getId() + " Değer Adı: " + sampleValue.getName());

} }

### Üst Kategori Bilgisini Gösterme (GetParentCategory)

Kodu verilen kategorinin birinci seviye üst kategorilerine ulaşmak için bu metot kullanılmalıdır. İkinci seviye üst kategorilere ulaşmak için (Örn. “Deri ayakkabı -> Ayakkabı -> Giysi” kategori ağacında “Giysi “ bilgisi) birinci seviye üst kategorinin (Örn. Ayakkabı) kodu verilerek tekrar servis çağırılmalıdır.

Sorgulanan kategori sistemde bulunamazsa ‘kategori bulunamadı’ hatası alınır. Eğer ilgili kategori herhangi bir üst kategoriye sahip değilse ”parentCategory” bilgisi response içerisinde yer almaz.

**GetParentCategoryRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| **categoryId** | Alt kategori id bilgisi |

**\*Bold alanlar zorunlu alanlardır.**

**GetParentCategoryResponse**

|  |  |
| --- | --- |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| category.id | Alt kategori id bilgisi |
| category.name | Kategori Adı |
| category.parentCategory.id | Bir üst kategori id bilgisi |
| category.parentCategory.name | Bir üst kategori adı |

**GetParentCategory Örnek Çağrı**

**Request**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Header/>

<soapenv:Body>

<sch:GetParentCategoryRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<categoryId>1002806</categoryId>

</sch:GetParentCategoryRequest>

</soapenv:Body></soapenv:Envelope>

***Response***

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:ParentCategoryResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<category>

<id>1002806</id>

<name>Tektaş Yüzük</name>

<parentCategory>

<id>1002799</id>

<name>Pırlanta Yüzük</name>

</parentCategory>

</category>

</ns3:ParentCategoryResponse>

</env:Body>

</env:Envelope>

**GetParentCategory Örnek Çağrı (Java)**

public static void main(String[] args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 long categoryIdVal = 1001886;  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 GetParentCategoryRequest request = new GetParentCategoryRequest();  
 request.setAuth(authentication);  
 request.setCategoryId(categoryIdVal);  
   
 CategoryServicePort port = new CategoryServicePortService().getCategoryServicePortSoap11();  
 GetParentCategoryResponse parentCategoryResponse = port.getParentCategory(request);  
 System.out.println("Parent category of \"" + parentCategoryResponse.getCategory().getName() + "\" is: " +  
 parentCategoryResponse.getCategory().getParentCategory().getName() + " " +  
 parentCategoryResponse.getCategory().getParentCategory().getId());  
}

### Alt Kategori Bilgilerini Listeleme (GetSubCategories)

Kodu verilen kategorinin birinci seviye alt kategorilerine ulaşmak için bu metot kullanılmalıdır. İkinci seviye alt kategorilere ulaşmak için (Örn. “Giysi -> Ayakkabı -> Deri ayakkabı” kategori ağacında “Deri ayakkabı” bilgisi) birinci Seviye alt kategorinin (Örn. Ayakkabı) kodu verilerek tekrar servis çağırılmalıdır.

Sorgulanan kategori sistemde bulunamazsa hata alınır. Eğer ilgili kategori herhangi bir alt kategoriye sahip değilse response bilgisi boş döner.

**GetSubCategoriesRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| **categoryId** | Alt kategori id bilgisi |

**\*Bold alanlar zorunlu alanlardır**

**GetParentCategoryResponse**

|  |  |
| --- | --- |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| category.id | Alt kategori id bilgisi |
| category.name | Kategori Adı |
| subCategoryList.subCategory.id | Bir alt kategori id bilgisi |
| subCategoryList.subCategory.name | Bir alt kategori adı |

**GetSubCategory Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetSubCategoriesRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<categoryId>1002720</categoryId>

</sch:GetSubCategoriesRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetSubCategoriesResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<category>

<id>1002720</id>

<name>Taşsız Bileklik</name>

</category>

</ns3:GetSubCategoriesResponse>

</env:Body>

</env:Envelope>

**GetSubCategory Örnek Çağrı (Java)**

public static void main(String[] args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 long categoryIdValue = 1002720L;  
   
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 GetSubCategoriesRequest request = new GetSubCategoriesRequest();  
 request.setAuth(authentication);  
 request.setCategoryId(categoryIdValue);  
   
 CategoryServicePort port = new CategoryServicePortService().getCategoryServicePortSoap11();  
 GetSubCategoriesResponse subCategoryResponse = port.getSubCategories(request);  
 List<SubCategoryData> subCategoryList = subCategoryResponse.getCategory();  
 System.out.println("Names of sub categories are: ");  
 for (SubCategoryData sampleData: subCategoryList  
 ) {  
 System.out.println(sampleData.getId() + " " + sampleData.getName()); }}

### Üst Kategori Bilgileri (GetTopLevelCategories)

n11 üzerinde tanımlanmış tüm üst seviye kategorileri görmek için bu metot kullanılır. Cevap olarak n11 kategori kodu ve kategori adı döner.

**GetTopLevelCategoriesRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |

**\*Bold alanlar zorunlu alanlardır.**

**GetTopLevelCategoriesResponse**

|  |  |
| --- | --- |
| categories.category.id | Kategorinin n11 tarafındaki kodu |
| categories.category.name | Kategori adı |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |

**GetTopLevelCategories Örnek Çağrı**

**Request**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<soapenv:Body>

<sch:GetTopLevelCategoriesRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

</sch:GetTopLevelCategoriesRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetTopLevelCategoriesResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<categories>

<category>

<id>1</id>

<name>Kategori 1</name>

</category>

<category>

<id>2</id>

<name>Kategori 2</name>

</category>

<category>

<id>3</id>

<name>Kategori 3</name>

</category>

<category>

<id>4</id>

<name>Kategori 4</name>

</category>

</categories>

</ns3:GetTopLevelCategoriesResponse>

</env:Body>

</env:Envelope>

**GetTopLevelCategories Örnek Çağrı (Java)**

public static void main (String [] args ) {  
   
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 GetTopLevelCategoriesRequest request = new GetTopLevelCategoriesRequest();  
 request.setAuth(authentication);  
  
 CategoryServicePort port = new CategoryServicePortService().getCategoryServicePortSoap11();  
 GetTopLevelCategoriesResponse topTopLevelCategoriesResponse = port.getTopLevelCategories(request);  
 List<SubCategory> categoryList = topTopLevelCategoriesResponse.getCategoryList().getCategory();  
  
 }

## 3.2 Şehir Servisleri (CityService)

Şehir servisleri için WSDL adresi: <https://api.n11.com/ws/CityService.wsdl>

### Şehirleri Listeleme (GetCities)

Sistemde kayıtlı şehirlerin kodları ve plaka numaraları ile birlikte listelenmesi için bu metot kullanılır. Adres ile ilgili işlem yapmak istendiği zaman bu servis aracılığı ile elde edilen şehir kodları kullanılır.

Genel kullanıma açık bir servis olduğu için servisin kullanımı sırasında herhangi bir güvenlik kontrolü yapılmamaktadır.

**GetCitiesResponse**

|  |  |
| --- | --- |
| Result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| cities.city.cityCode | Şehrin plaka kodu |
| cities.city.cityId | Şehrin n11 tarafındaki kodu |
| cities.city.cityName | Şehir İsmi |

**GetCities Örnek Çağrı**

**Request**

<soapenv:Envelopexmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetCitiesRequest/>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<soapenv:Envelopexmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/”>

<env:Header/>

<env:Body>

<ns3:GetCitiesResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<cities>

<city>

<cityCode>3</cityCode>

<cityId>424</cityId>

<cityName>Afyon</cityName>

</city>

<city>

<cityCode>1</cityCode>

<cityId>423</cityId>

<cityName>Adıyaman</cityName>

</city>

<city>

<cityCode>1</cityCode>

<cityId>422</cityId>

<cityName>Adana</cityName>

</city>

<city>

<cityCode>4</cityCode>

<cityId>425</cityId>

<cityName>Ağrı</cityName>

</city>

…

</cities>

</ns3:GetCitiesResponse>

</env:Body>

</env:Envelope>

***GetCities Örnek Çağrı (Java)***

public static void main (String [] args) {

GetCitiesRequest request = new GetCitiesRequest();

CityServicePort port = new CityServicePortService().getCityServicePortSoap11();

GetCitiesResponse getCitiesResponse = port.getCities(request);

List<CityData> cityList = getCitiesResponse.getCities().getCity();

System.out.println("Cities are:");

for (CityData sampleCity:cityList

) {

System.out.println(sampleCity.getCityId() + " " + sampleCity.getCityName() + " ("+ sampleCity.getCityCode() + ")");

}

}

### Şehir Bilgisi Gösterme (GetCity)

Şehir plaka numarası verilen şehrin sistemde kayıtlı olan kodunu ve hangi şehir olduğunu öğrenmek için bu metot kullanılmalıdır.

Genel kullanıma açık bir servis olduğu için servisin kullanımı sırasında herhangi bir güvenlik kontrolü yapılmamaktadır.

Sorgulanan şehir sistemde bulunamazsa ‘şehir bulunamadı’ hatası alınır.

**GetCityRequest**

|  |  |
| --- | --- |
| **cityCode** | Şehrin plaka kodu |

**\*Bold alanlar zorunlu alanlardır.**

**GetCityResponse**

|  |  |
| --- | --- |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| city.cityCode | Şehir kodu |
| city.cityId | Şehre n11 tarafından verilen id |
| city.cityName | Şehir adı |

**GetCity Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetCityRequest>

<cityCode>05</cityCode>

</sch:GetCityRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetCityResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<city>

<cityCode>5</cityCode>

<cityId>2505</cityId>

<cityName>Amasya</cityName>

</city>

</ns3:GetCityResponse>

</env:Body>

</env:Envelope>

**GetCity Örnek Çağrı (JAVA)**

public static void main (String [] args) {  
 int cityCodeValue = 26;  
 GetCityRequest request = new GetCityRequest();  
 request.setCityCode(BigInteger.valueOf(cityCodeValue));  
 CityServicePort port = new CityServicePortService().getCityServicePortSoap11();  
 GetCityResponse cityResponse = port.getCity(request);  
 CityData sampleCity = cityResponse.getCity();  
 System.out.println("The city is: " + sampleCity.getCityId() + " " + sampleCity.getCityName() + " ("+ sampleCity.getCityCode() + ")" );  
}

### İlçe Bilgilerinin Gösterimi (GetDistrict)

Plaka kodu verilen şehre ait ilçelerinin listelenmesi için bu metot kullanılmalıdır. İlçe kodu adres ekleme/güncelleme işlemlerinde kullanılmaktadır.

Genel kullanıma açık bir servis olduğu için servisin kullanımı sırasında herhangi bir güvenlik kontrolü yapılmamaktadır.

Sorgulanan şehir sistemde bulunamazsa ‘şehir bulunamadı’ hatası alınır.

**GetDistrictRequest**

|  |  |
| --- | --- |
| **cityCode** | Şehrin plaka kodu |

**\*Bold alanlar zorunlu alanlardır.**

**GetDistrictResponse**

|  |  |
| --- | --- |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| districts.district.id | Şehrin plaka kodu |
| districts.district.name | İlçenin n11 tarafındaki kodu |

**GetDistrict Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetDistrictRequest>

<cityCode>05</cityCode>

</sch:GetDistrictRequest>

</soapenv:Body>

</soapenv:Envelope>

***Response***

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetDistrictResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<districts>

<district>

<id>22251</id>

<name>Gümüşhacıköy</name>

</district>

<district>

<id>22252</id>

<name>Merzifon</name>

</district>

<district>

<id>22711</id>

<name>Göynücek</name>

</district>

<district>

<id>22712</id>

<name>Merkez</name>

</district>

<district>

<id>22713</id>

<name>Suluova</name>

</district>

<district>

<id>22714</id>

<name>Taşova</name>

</district>

<district>

<id>22487</id>

<name>Hamamözü</name>

</district>

</districts>

</ns3:GetDistrictResponse>

</env:Body>

</env:Envelope>

**GetDistrict Örnek Çağrı (Java)**

public static void main(String[] args) {  
 int cityCodeValue = 05;  
   
 GetDistrictRequest request = new GetDistrictRequest();  
 request.setCityCode(BigInteger.*valueOf*(cityCodeValue));  
   
 CityServicePort port = new CityServicePortService().getCityServicePortSoap11();  
 GetDistrictResponse getDistrictResponse = port.getDistrict(request);  
   
 List<DistrictData> districtList = getDistrictResponse.getDistricts().getDistrict();  
  
 for (DistrictData sampleDistrict:districtList  
 ) {  
 System.*out*.println(sampleDistrict.getId() + " " + sampleDistrict.getName());  
 }  
}

### Semt/Mahalle Bilgilerinin Gösterimi (GetNeighborhoods)

İlçe kodu verilen semt/mahallelerin listelenmesi için bu metot kullanılmalıdır.

Genel kullanıma açık bir servis olduğu için servisin kullanımı sırasında herhangi bir güvenlik kontrolü yapılmamaktadır.

Sorgulanan ilçe sistemde bulunamazsa ‘ilçe bulunamadı’ hatası alınır.

**GetNeighborhoodsRequest**

|  |  |
| --- | --- |
| **districtId** | İlçe kodu |

**\*Bold alanlar zorunlu alanlardır.**

**GetNeighborhoodsResponse**

|  |  |
| --- | --- |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| Neighborhoods.neighborhood.id | Semt/mahalle kodu |
| Neighborhoods.neighborhood.name | Semt/mahalle’nin n11 tarafındaki kodu |

**GetNeighborhoods Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetNeighborhoodsRequest>

<districtId>26633</districtId>

</sch:GetNeighborhoodsRequest>

</soapenv:Body>

</soapenv:Envelope>

***Response***

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetNeighborhoodsResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<neighborhoods>

<neighborhood>

<id>354742</id>

<name>Ayazağa</name>

</neighborhood>

<neighborhood>

<id>308456</id>

<name>Bahçeköy Merkez</name>

</neighborhood>

<neighborhood>

<id>326752</id>

<name>Bahçeköy Yeni</name>

</neighborhood>

<neighborhood>

<id>354743</id>

<name>Baltalimanı</name>

</neighborhood>

...

<neighborhood>

<id>290399</id>

<name>İstinye</name>

</neighborhood>

</neighborhoods>

</ns3:GetNeighborhoodsResponse>

</env:Body>

</env:Envelope>

**GetNeighborhoods Örnek Çağrı (Java)**

public static void main(String[] args) {

long neighborhoodCode = 26633;

GetNeighborhoodsRequest request = new GetNeighborhoodsRequest();

request.setDistrictId(neighborhoodCode);

CityServicePort port = new CityServicePortService().getCityServicePortSoap11();

GetNeighborhoodsResponse getNeighborhoodsResponse = port.getNeighborhoods(request);

List<NeighborhoodData> neighborhoodDataList = getNeighborhoodsResponse.getNeighborhoods().getNeighborhood();

for (NeighborhoodData sampleNeigborhood : neighborhoodDataList

) {

System.out.println(sampleNeigborhood.getId() + " " + sampleNeigborhood.getName());

}

}

## 3.3 Ürün Servisleri (ProductService)

Ürün servisleri için WSDL adresi:

<https://api.n11.com/ws/ProductService.wsdl>

### N11 Ürün ID si ile Ürün Bilgilerini Alma (GetProductByProductId)

N11 ürün ID sini kullanarak sistemde kayıtlı olan ürünün bilgilerini getirir.

**GetProductByIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Ürün N11 ID si* |

**\*Bold alanlar zorunlu alanlardır.**

**GetProductByIdResponse**

|  |  |
| --- | --- |
| result | [Bkz: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| *product.currencyAmount* | *Görüntülenen ürün fiyatının para birimi tutarı* |
| *product.currencyType* | *Görüntülenen ürün fiyatının para birimi cinsi (USD/EUR/TL)* |
| *product.displayPrice* | *Görüntülenen ürün fiyatı (Ürünün indirimler sonucu tanımlanan son fiyat hali)* |
| *product.id* | *Ürünün N11 ID si* |
| *product.price* | *Ürünün liste fiyatı* |
| *product.productSellerCode* | *Mağaza ürün kodu* |
| *product.saleStatus* | [*Bkz: Ürün Satış Durumu*](#_Ürün_Satış_Durumu) |
| *product.approvalStatus* | [*Bkz: Ürün Onay Durumu*](#_Ürün_Onay_Durumu) |
| *product.subtitle* | *Ürün alt başlığı* |
| *product.title* | *Ürün başlığı* |
| *product.category.fullName* | *Ürün kategori bilgisi tüm kategori alt başlıkları ile* |
| *product.category.id* | *Ürün kategori id* |
| *product.category.name* | *Ürün kategori adı* |
| *product.images.image.order* | *Resim öncelik numarası* |
| *product.images.image.url* | *Resim url adresi* |
| *product.preparingDay* | *Ürünün kargoya verilme süresi (gün olarak)* |
| *product.productCondition* | [*Bkz: Ürün Durumu*](#_Ürün_Durumu_(Product) |
| *product.saleEndDate* | *Ürünün son satış tarihi* |
| *product.saleStartDate* | *Ürünün satışa başladığı tarih* |
| *product.shipmentTemplate* | *Ürün için belirlenen teslimat şablonu* |
| *product.discount* | [*Bkz: Ürün İndirim Bilgisi*](#_Ürün_İndirim_Bilgisi_1) |
| *product.stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *product.stockItems.stockItem.quantity* | *Ürün stok miktarı* |
| *product.stockItems.stockItem.version* | *Ürün stok versiyon numarası* |
| *product.stockItems.stockItem.currencyAmount* | *Ürün stok biriminin para birimi cinsinden tutarı* |
| *product.stockItems.stockItem.displayPrice* | *Ürün stok biriminin görüntülenen ürün fiyatı (Ürünün satıcı indirimi sonucu tanımlanan son fiyat hali)* |
| *product.stockItems.stockItem.optionPrice* | *Ürün stok biriminin liste fiyatı* |
| *product.stockItems.stockItem.sellerStockCode* | *Ürün stok mağaza kodu* |
| product.stockItems.stockItem.bundle | *Birlikte satış yapılan ürünler* |
| product.stockItems.stockItem.mpn | *Ürünün üretici parça numarası* |
| product.stockItems.stockItem.gtin | *Ürünün global ticari öğe numarası* |
| *product.stockItems.stockItem.attributes.attribute.name* | *Stok tanımının olduğu seçenek adı* |
| *product.stockItems.stockItem.attributes.attribute.value* | *Stok tanımının olduğu seçenek değeri* |

**GetProductByProductId Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetProductByProductIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productId>16685311</productId>

</sch:GetProductByProductIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetProductByProductIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>3000.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>3000.00</displayPrice>

<id>16685311</id>

<price>3000.00</price>

<productSellerCode>Deneme1234567</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>3000.00</currencyAmount>

<displayPrice>3000.00</displayPrice>

<gtin>0190198066474</gtin>

<optionPrice>3000.00</optionPrice>

<sellerStockCode>StokKodu1234567890-2</sellerStockCode>

<attributes>

<attribute>

<id>354091250</id>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<id>121876345662</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

<stockItem>

<currencyAmount>3000.00</currencyAmount>

<displayPrice>3000.00</displayPrice>

<gtin>0190198066473</gtin>

<optionPrice>3000.00</optionPrice>

<sellerStockCode>StokKodu1234567890-1</sellerStockCode>

<attributes>

<attribute>

<id>354091250</id>

<name>Renk</name>

<value>Mavi</value>

</attribute>

</attributes>

<id>121876345663</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Lorem ipsum dolor sit amet</subtitle>

<title>Lorem Ipsum</title>

<attributes>

<attribute>

<id>354081904</id>

<name>Marka</name>

<value>Apple</value>

</attribute>

</attributes>

<category>

<fullName>Telefon &amp; Aksesuarları > Cep Telefonu</fullName>

<id>1000476</id>

<name>Cep Telefonu</name>

</category>

<description>Hello World!</description>

<images>

<image>

<order>1</order>

<url>http://dev.ic.n11.com/a1/org/elektronik/cep-telefonu/lorem-ipsum\_\_0163849896557450.jpg</url>

</image>

</images>

<preparingDay>1</preparingDay>

<productCondition>1</productCondition>

<productionDate>01/12/2016</productionDate>

<saleEndDate>26/12/2017</saleEndDate>

<saleStartDate>26/12/2016</saleStartDate>

<shipmentTemplate>AGT</shipmentTemplate>

<specialProductInfoList/>

</product>

</ns3:GetProductByProductIdResponse>

</env:Body>

</env:Envelope>

**GetProductByProductId Örnek Çağrı (JAVA)**

public static void main(String[] args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 long productIdValue = 125661592L;  
   
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 GetProductByProductIdRequest request = new GetProductByProductIdRequest();  
 request.setAuth(authentication);  
 request.setProductId(productIdValue);  
   
 ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();  
 GetProductByProductIdResponse response = port.getProductByProductId(request);  
   
 Product sampleProduct = response.getProduct();  
 System.*out*.println("Name of product is: " + sampleProduct.getTitle());  
}

### Mağaza Ürün Kodu ile Ürün Bilgilerini Alma (GetProductBySellerCode)

Mağaza ürün kodunu kullanarak sistemde kayıtlı olan ürünün bilgilerini getirir.

**GetProductBySellerCodeRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***sellerCode*** | *Ürün mağaza kodu* |

**\*Bold alanlar zorunlu alanlardır.**

**GetProductBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product.currencyAmount* | *Görüntülenen ürün fiyatının para birimi tutarı* |
| *product.currencyType* | *Görüntülenen ürün fiyatının para birimi cinsi (USD/EUR/TL)* |
| *product.displayPrice* | *Görüntülenen ürün fiyatı (Ürünün indirimler sonucu tanımlanan son fiyat hali)* |
| *product.id* | *Ürünün N11 ID si* |
| *product.price* | *Ürünün Liste fiyatı* |
| *product.productSellerCode* | *Mağaza ürün kodu* |
| *product.saleStatus* | [*Bkz: Ürün Satış Durumu*](#_Ürün_Satış_Durumu_1) |
| *Product.approvalStatus* | [*Bkz: Ürün Onay Durumu*](#_Ürün_Onay_Durumu_1) |
| *product.subtitle* | *Ürün alt başlığı* |
| *product.title* | *Ürün başlığı* |
| *product.category.fullName* | *Ürün kategori bilgisi tüm kategori alt başlıkları ile* |
| *product.category.id* | *Ürün kategori id* |
| *product.category.name* | *Ürün kategori adı* |
| *product.images.image.order* | *Resim öncelik numarası* |
| *product.images.image.url* | *Resim url adresi* |
| *product.preparingDay* | *Ürünün kargoya verilme süresi (gün olarak)* |
| *product.productCondition* | [*Bkz: Ürün Durumu*](#_Ürün_Durumu_(Product) |
| *product.saleEndDate* | *Ürünün son satış tarihi* |
| *product.saleStartDate* | *Ürünün satışa başladığı tarih* |
| *product.shipmentTemplate* | *Ürün için belirlenen teslimat şablonu* |
| *product.discount* | [*Bkz: Ürün İndirim Bilgisi*](#_Ürün_İndirim_Bilgisi_1) |
| *product.stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *product.stockItems.stockItem.quantity* | *Ürün stok miktarı* |
| *product.stockItems.stockItem.version* | *Ürün stok versiyon numarası* |
| *product.stockItems.stockItem.currencyAmount* | *Ürün stok biriminin para birimi cinsinden tutarı* |
| *product.stockItems.stockItem.displayPrice* | *Ürün stok biriminin görüntülenen ürün fiyatı (Ürünün satıcı indirimi sonucu tanımlanan son fiyat hali)* |
| *product.stockItems.stockItem.optionPrice* | *Ürün stok biriminin liste fiyatı* |
| *product.stockItems.stockItem.sellerStockCode* | *Ürün stok mağaza kodu* |
| product.stockItems.stockItem.bundle | *Birlikte satış yapılan ürünler* |
| product.stockItems.stockItem.mpn | *Ürünün üretici parça numarası* |
| product.stockItems.stockItem.gtin | *Ürünün global ticari öğe numarası* |
| *product.stockItems.stockItem.attributes.attribute.name* | *Stok tanımının olduğu seçenek adı* |
| *product.stockItems.stockItem.attributes.attribute.value* | *Stok tanımının olduğu seçenek değeri* |

**GetProductBySellerCode Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetProductBySellerCodeRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<sellerCode>Deneme1234567</sellerCode>

</sch:GetProductBySellerCodeRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetProductBySellerCodeResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>3000.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>3000.00</displayPrice>

<id>16685311</id>

<price>3000.00</price>

<productSellerCode>Deneme1234567</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>3000.00</currencyAmount>

<displayPrice>3000.00</displayPrice>

<gtin>0190198066474</gtin>

<optionPrice>3000.00</optionPrice>

<sellerStockCode>StokKodu1234567890-2</sellerStockCode>

<attributes>

<attribute>

<id>354091250</id>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<id>121876345662</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

<stockItem>

<currencyAmount>3000.00</currencyAmount>

<displayPrice>3000.00</displayPrice>

<gtin>0190198066473</gtin>

<optionPrice>3000.00</optionPrice>

<sellerStockCode>StokKodu1234567890-1</sellerStockCode>

<attributes>

<attribute>

<id>354091250</id>

<name>Renk</name>

<value>Mavi</value>

</attribute>

</attributes>

<id>121876345663</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Lorem ipsum dolor sit amet</subtitle>

<title>Lorem Ipsum</title>

<attributes>

<attribute>

<id>354081904</id>

<name>Marka</name>

<value>Apple</value>

</attribute>

</attributes>

<category>

<fullName>Telefon &amp; Aksesuarları > Cep Telefonu</fullName>

<id>1000476</id>

<name>Cep Telefonu</name>

</category>

<description>Hello World!</description>

<images>

<image>

<order>1</order>

<url>http://dev.ic.n11.com/a1/org/elektronik/cep-telefonu/lorem-ipsum\_\_0163849896557450.jpg</url>

</image>

</images>

<preparingDay>1</preparingDay>

<productCondition>1</productCondition>

<productionDate>01/12/2016</productionDate>

<saleEndDate>26/12/2017</saleEndDate>

<saleStartDate>26/12/2016</saleStartDate>

<shipmentTemplate>AGT</shipmentTemplate>

<specialProductInfoList/>

</product>

</ns3:GetProductBySellerCodeResponse>

</env:Body>

</env:Envelope>

**GetProductByProductSellerCode Örnek Çağrı (JAVA)**

public static void main(String[] args) {  
  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 String strSellerCode = "8690536804658";  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
   
 GetProductBySellerCodeRequest request = new GetProductBySellerCodeRequest();  
 request.setAuth(authentication);  
 request.setSellerCode(strSellerCode);  
   
 ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();  
 GetProductBySellerCodeResponse response = port.getProductBySellerCode(request);  
}

### Ürün Listeleme (GetProductList)

Sistemde ürünleri listelemek için kullanılır. Metot ürün ile ilgili özet bilgileri getirmektedir. Bu metottan dönen “**productSellerCode**” bilgisini kullanarak “**GetProductBySellerCode**” metodu veya “**id**” bilgisini kullanarak “**GetProductById**” metodu kullanılarak ürünün detaylı bilgilerine ulaşılabilir.

Sonuçlar, sayfalama bilgisi ile beraber döner. “**itemsPerPage**” parametresi ile her sayfada listelenecek ürün adedi, “**currentPage**” parametresi ile de listelenmek istenen sayfanın numarası kontrol edilebilir. Herhangi bir değer girilmediği takdirde, her sayfada **100** adet ürün listelenir ve birinci sayfa listelenir.

“**pagingData**” içinde toplam ürün adedi, sayfa sayısı, mevcut sayfa ve sayfa başına düşen ürün adedi bilgileri kullanıcılara verilir.

**GetProductListRequest**

|  |  |
| --- | --- |
| **auth** | [Bkz: Yetkilendirme](#_Yetkilendirme_(Authentication)) |
| pagingData | [Bkz: Sayfalandırma](#_Sayfalandırma_(Pagination)) |

**\*Bold alanlar zorunlu alanlardır.**

**GetProductListResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*Bkz: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**GetProductList Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetProductListRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

</pagingData>

</sch:GetProductListRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetProductListResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<products>

<product>

<currencyAmount>3000.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>3000.00</displayPrice>

<id>16685311</id>

<price>3000.00</price>

<productSellerCode>Deneme1234567</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>3000.00</currencyAmount>

<displayPrice>3000.00</displayPrice>

<gtin>0190198066474</gtin>

<optionPrice>3000.00</optionPrice>

<sellerStockCode>StokKodu1234567890-2</sellerStockCode>

<attributes>

<attribute>

<id>354091250</id>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<id>121876345662</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

<stockItem>

<currencyAmount>3000.00</currencyAmount>

<displayPrice>3000.00</displayPrice>

<gtin>0190198066473</gtin>

<optionPrice>3000.00</optionPrice>

<sellerStockCode>StokKodu1234567890-1</sellerStockCode>

<attributes>

<attribute>

<id>354091250</id>

<name>Renk</name>

<value>Mavi</value>

</attribute>

</attributes>

<id>121876345663</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Lorem ipsum dolor sit amet</subtitle>

<title>Lorem Ipsum</title>

</product>

...

</products>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

<totalCount>9</totalCount>

<pageCount>1</pageCount>

</pagingData>

</ns3:GetProductListResponse>

</env:Body>

</env:Envelope>

***GetProductList Örnek Çağrı (JAVA)***

public static void main (String []args) {  
   
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 int currentPageValue = 0;  
 int pageSizeValue = 3;  
   
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 RequestPagingData requestPagingData = new RequestPagingData();  
 requestPagingData.setCurrentPage(currentPageValue);  
 requestPagingData.setPageSize(pageSizeValue);  
  
 GetProductListRequest getProductListRequest = new GetProductListRequest();  
 getProductListRequest.setAuth(authentication);  
 getProductListRequest.setPagingData(requestPagingData);  
  
 ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();  
 GetProductListResponse response = port.getProductList(getProductListRequest);  
 List<ProductBasic> productList = response.getProducts().getProduct();  
  
 for (ProductBasic sampleProduct:productList  
 ) {  
 System.out.println("Title of product is : " + sampleProduct.getTitle());  
 }  
}

### Ürün Kaydetme (Save Product)

Yeni ürün oluşturmak veya mevcut ürünü güncellemek için kullanılır. Ürün kodu “**productSellerCode**” adıyla veirlen data bu mağaza için bir kayıtlı bir ürünü gösteriyorsa bu ürün güncellenir, aksi halde yeni bir ürün oluşturulur.

“**stockItems**” içinde hem seçenekleri olan ürünlerin (numara, beden gibi), hem de seçeneği olmayan ürünlerin stok girişi yapılabilir. Seçeneği olmayan ürünler için “**attributes**” alanını boş bırakılabilir.

“**category.id**” içinde ürünün dahil olacağı kategori N11 ID bilgisi yer almalıdır, kategori listesi **CategoryService** **GetParentCategory** ile başlayıp **GetSubCategories** metodları ile alınabilir, ürün dahil olacağı kategori en alt seviyede olmalıdır.

"templateName" içinde sistemde bu mağaza için tanımlı olan bir teslimat şablonu kullanılmalıdır. Teslimat şablonları tanımlamak için Satış Ofisi ekranlarından Hesabım > Teslimat Bilgileri menü adımları takip edilerek manuel olarak oluşturulabilir veya **ShipmentService** içinden **createOrUpdateShipmentTemplate** metodu kullanılabilir. Seyahat kategori ürünleri için özel açıklama bölümünün girileceği alan specialProductInfoList’dir.

**SaveProductRequest**

|  |  |
| --- | --- |
| Auth | [Bkz:: Yetkilendirme](#_Yetkilendirme_(Authentication)) |
| **product.productSellerCode** | Ürün Mağaza Kodu |
| **product.title** | [Bkz:: title](#_Ürün_Başlığı_(Title)) |
| **product.subtitle** | [Bkz:: subtitle](#_Ürün_Altbaşlığı_(Subtitle)) |
| **product.description** | Ürün açıklama bilgileri (HTML olabilir) |
| product.attributes.attribute | Ürün özelliklerinin adı ve değeri girilen alan |
| **product.category.id** | Ürün kategori numarası |
| **product.price** | Ürün baz fiyatı |
| **product.currencyType** | Ürün liste fiyatı para birimi |
| **product.images.image.url** | Ürün resmi URL adresi |
| **product.images.image.order** | Ürün resmi gösterilme sırası |
| product.saleStartDate | Ürün satış başlangıç tarihi (dd/MM/yyyy), boş gönderilirse ürün aynı gün satışa çıkar |
| product.saleEndDate | Ürün satış bitiş tarihi (dd/MM/yyyy), boş gönderilirse çok ileri bir tarihe atanır |
| product.productionDate | Ürün üretim tarihi (dd/MM/yyyy) |
| product.expirationDate | Ürün son kullanma tarihi (dd/MM/yyyy) |
| **product.productCondition** | [Bkz: Ürün Durumu](#_Ürün_Durumu_(Product_1) |
| **product.preparingDay** | Ürün kargoya verilme süresi (gün olarak) |
| product.discount | [Bkz: Ürün İndirim Bilgisi](#_Ürün_İndirim_Bilgisi_2) |
| **product.shipmentTemplate** | Teslimat Şablon Adı |
| **product.stockItems.stockItem.quantity** | Stok miktarı |
| product.stockItems.stockItem.sellerStockCode | Mağaza ürün stok kodu |
| product.stockItems.stockItem.attributes.attribute.name | Eğer ürünün seçenekleri varsa bu bilgiler girilir, isim olarak kendi belirlediğiniz bir değerleri kullanabilirsiniz. |
| product.stockItems.stockItem.attributes.attribute.value | Eğer ürünün seçenekleri varsa bu bilgiler girilir, değer olarak kendi belirlediğiniz bir değerleri kullanabilirsiniz. |
| product.stockItems.stockItem.optionPrice | Ürün stok biriminin liste fiyatı |
| product.stockItems.stockItem.bundle | *Birlikte satış yapılan ürünler* |
| product.stockItems.stockItem.mpn | *Ürünün üretici parça numarası* |
| product.stockItems.stockItem.gtin | *Ürünün global ticari öğe numarası* |
| specialProductInfoList.specialProductInf.key | Seyahat kategori ürün açıklama bilgilerinin girildiği alan. Bu Alana; TurProgrami, IptalIadeKosullari,EkHizmetler bilgileri girilerek açıklama oluşturulur |
| specialProductInfoList.specialProductInf.value | Key’e karşılık gelen html verisi |

**\*Bold alanlar zorunlu alanlardır.**

**SaveProductResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*Bkz: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**SaveProduct Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:SaveProductRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<product>

<productSellerCode>Deneme1234567</productSellerCode>

<title>Lorem Ipsum</title>

<subtitle>Lorem ipsum dolor sit amet</subtitle>

<description>Hello World!</description>

<category>

<id>1000476</id>

</category>

<specialProductInfoList/>

<price>3000</price>

<currencyType>1</currencyType>

<images>

<!--1 or more repetitions:-->

<image>

<url>https://www.google.com/logos/doodles/2016/bahrain-national-day-2016-6221988579246080-hp2x.jpg</url>

<order>1</order>

</image>

</images>

<approvalStatus>1</approvalStatus>

<attributes>

<!--1 or more repetitions:-->

<attribute>

<name>Marka</name>

<value>Apple</value>

</attribute>

<attribute>

<name>Renk</name>

<value>Mavi</value>

</attribute>

</attributes>

<saleStartDate>26/12/2016</saleStartDate>

<saleEndDate>26/12/2017</saleEndDate>

<productionDate>01/12/2016</productionDate>

<expirationDate/>

<productCondition>1</productCondition>

<preparingDay>1</preparingDay>

<discount>

<startDate/>

<endDate/>

<type/>

<value/>

</discount>

<shipmentTemplate>AGT</shipmentTemplate>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<bundle/>

<mpn/>

<gtin>0190198066473</gtin>

<quantity>5</quantity>

<sellerStockCode>StokKodu1234567890-1</sellerStockCode>

<attributes>

<!--1 or more repetitions:-->

<attribute>

<name>Renk</name>

<value>Mavi</value>

</attribute>

</attributes>

<optionPrice/>

</stockItem>

<stockItem>

<bundle/>

<mpn/>

<gtin>0190198066474</gtin>

<quantity>5</quantity>

<sellerStockCode>StokKodu1234567890-2</sellerStockCode>

<attributes>

<!--1 or more repetitions:-->

<attribute>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<optionPrice/>

</stockItem>

</stockItems>

</product>

</sch:SaveProductRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:SaveProductResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

<errorMessage/>

</result>

<product>

<currencyAmount>3000.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>3000.00</displayPrice>

<id>16685311</id>

<price>3000.00</price>

<productSellerCode>Deneme1234567</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>3000.00</currencyAmount>

<displayPrice>3000.00</displayPrice>

<gtin>0190198066474</gtin>

<mpn/>

<optionPrice>3000.00</optionPrice>

<sellerStockCode>StokKodu1234567890-2</sellerStockCode>

<attributes>

<attribute>

<id>354091250</id>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<id>121876345662</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

<stockItem>

<currencyAmount>3000.00</currencyAmount>

<displayPrice>3000.00</displayPrice>

<gtin>0190198066473</gtin>

<mpn/>

<optionPrice>3000.00</optionPrice>

<sellerStockCode>StokKodu1234567890-1</sellerStockCode>

<attributes>

<attribute>

<id>354091250</id>

<name>Renk</name>

<value>Mavi</value>

</attribute>

</attributes>

<id>121876345663</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Lorem ipsum dolor sit amet</subtitle>

<title>Lorem Ipsum</title>

</product>

</ns3:SaveProductResponse>

</env:Body>

</env:Envelope>

**SaveProduct Örnek Çağrı (JAVA)**

public static void main (String []args) {

String strAppKey = "64155786-da91-4204-8735-443d17acf808";

String strAppSecret = "\*\*\*";

String strUrl = "https://www.google.com/logos/doodles/2016/bahrain-national-day-2016-6221988579246080-hp2x.jpg";

String strSellerStockCode = "MaviKod-APIDeneme432100000000";

String strSellerStockCode1 = "KırmızıKod-APIDeneme4321000000000";

String strAttributeName = "Marka";

String strAttributeValue = "Apple";

String strSkuAttributeKey = "Renk";

String strSkuAttributeValue = "Mavi";

String strSkuAttributeValue1 = "Kırmızı";

String strProductTitle = "Lorem ipsum";

String strProductSubtitle = "Lorem ipsum dolor sit amet";

String strProductSellerCode = "APIDeneme432101000000000";

String strProductCondition = "1";

String strShipmentTemplate = "AGT";

String strProductDescription = "Hello World!";

String strGtin = "0190198066473";

String strGtin1 = "0190198066474";

int setOrderValue = 1;

int quantityValue = 10;

int quantityValue1 = 20;

int categoryIdValue = 592;

int priceValue = 50;

int currencyTypeValue = 1;

int approvalStatusValue = 1;

int preparingDayValue = 3;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

ProductImage productImage = new ProductImage();

ProductImageList productImageList = new ProductImageList();

productImage.setUrl(strUrl);

productImage.setOrder(BigInteger.valueOf(setOrderValue));

productImageList.getImage().add(productImage);

ProductAttributeRequestList skuAttributeRequestList = new ProductAttributeRequestList();

ProductAttributeRequest skuAttributeRequest = new ProductAttributeRequest();

skuAttributeRequest.setName(strSkuAttributeKey);

skuAttributeRequest.setValue(strSkuAttributeValue);

skuAttributeRequestList.getAttribute().add(skuAttributeRequest);

ProductAttributeRequestList skuAttributeRequestList1 = new ProductAttributeRequestList();

ProductAttributeRequest skuAttributeRequest1 = new ProductAttributeRequest();

skuAttributeRequest1.setName(strSkuAttributeKey);

skuAttributeRequest1.setValue(strSkuAttributeValue1);

skuAttributeRequestList1.getAttribute().add(skuAttributeRequest1);

ProductSkuRequestList stockItems = new ProductSkuRequestList();

ProductSkuRequest sku = new ProductSkuRequest();

sku.setSellerStockCode(strSellerStockCode);

sku.setAttributes(skuAttributeRequestList);

sku.setQuantity(BigInteger.valueOf(quantityValue));

sku.setGtin(strGtin);

ProductSkuRequest sku1 = new ProductSkuRequest();

sku1.setSellerStockCode(strSellerStockCode1);

sku1.setAttributes(skuAttributeRequestList1);

sku1.setQuantity(BigInteger.valueOf(quantityValue1));

sku1.setGtin(strGtin1);

stockItems.getStockItem().add(sku);

stockItems.getStockItem().add(sku1);

CategoryRequest categoryRequest = new CategoryRequest();

categoryRequest.setId(categoryIdValue);

ProductAttributeRequest productAttribute = new ProductAttributeRequest();

productAttribute.setName(strAttributeName);

productAttribute.setValue(strAttributeValue);

ProductAttributeRequestList productAttributeRequestList = new ProductAttributeRequestList();

productAttributeRequestList.getAttribute().add(productAttribute);

ProductRequest productRequest = new ProductRequest();

productRequest.setTitle(strProductTitle);

productRequest.setSubtitle(strProductSubtitle);

productRequest.setDescription(strProductDescription);

productRequest.setCategory(categoryRequest);

productRequest.setProductSellerCode(strProductSellerCode);

productRequest.setPrice(BigDecimal.valueOf(priceValue));

productRequest.setCurrencyType(BigInteger.valueOf(currencyTypeValue));

productRequest.setImages(productImageList);

productRequest.setApprovalStatus(BigInteger.valueOf(approvalStatusValue));

productRequest.setPreparingDay(BigInteger.valueOf(preparingDayValue));

productRequest.setStockItems(stockItems);

productRequest.setProductCondition(strProductCondition);

productRequest.setShipmentTemplate(strShipmentTemplate);

productRequest.setAttributes(productAttributeRequestList);

SaveProductRequest saveProductRequest = new SaveProductRequest();

saveProductRequest.setAuth(authentication);

saveProductRequest.setProduct(productRequest);

ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();

SaveProductResponse response = port.saveProduct(saveProductRequest);

System.out.println("Saving product " + response.getProduct().getId() + " is " + response.getResult().getStatus().getValue());

}

### Ürün Arama (SearchProducts)

Mağaza ürünlerini aramak için kullanılır.

**SearchProductsRequest**

|  |  |
| --- | --- |
| **Auth** | [Bkz: Yetkilendirme](#_Yetkilendirme_(Authentication)) |
| pagingData.currentPage | Arama sonrasında istenen sayfa |
| pagingData.pageSize | Bir sayfada gösterilecek madde sayısı |
| productSearch.name | Arama yapılmak istenen kelime |
| productSearch.saleDate.startDate | Ürünün satışa başladığı tarih |
| productSearch.saleDate.endDate | Ürünün satışının bittiği tarih |
| approvalStatus | [Bkz: ApprovalStatus](#_Onay_Durumu_(Approval) |

**\*Bold alanlar zorunlu alanlardır.**

**SearchProductsResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*Bkz: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**SearchProducts Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:SearchProductsRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<pagingData>

<currentPage>0</currentPage>

<pageSize>2</pageSize>

</pagingData>

<productSearch>

<name>Lorem</name>

<saleDate>

<startDate>20.09.2012</startDate>

<endDate>18.09.2016</endDate>

</saleDate>

<approvalStatus>1</approvalStatus>

</productSearch>

</sch:SearchProductsRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:SearchProductsResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<products>

<product>

<currencyAmount>50.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>50.00</displayPrice>

<id>126294855</id>

<price>50.00</price>

<productSellerCode>denemekod123</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>50.00</currencyAmount>

<displayPrice>50.00</displayPrice>

<optionPrice>50.00</optionPrice>

<sellerStockCode>deneme4321</sellerStockCode>

<attributes/>

<id>123495502203</id>

<quantity>10</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Lorem ipsum dolor sit amet</subtitle>

<title>Lorem ipsum</title>

</product>

</products>

<pagingData>

<currentPage>0</currentPage>

<pageSize>2</pageSize>

<totalCount>1</totalCount>

<pageCount>1</pageCount>

</pagingData>

</ns3:SearchProductsResponse>

</env:Body>

</env:Envelope>

**SearchProducts Örnek Çağrı (JAVA)**

public static void main (String [] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strStartDate = "20.09.2012";

String strEndDate = "18.09.2016";

String strProductName = "Lorem";

int currentPageValue = 0;

int pageSizeValue = 2;

ProductStatus productStatus = ProductStatus.ACTIVE;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

RequestPagingData requestPagingData = new RequestPagingData();

requestPagingData.setCurrentPage(currentPageValue);

requestPagingData.setPageSize(pageSizeValue);

DateRange dateRange = new DateRange();

dateRange.setStartDate(strStartDate);

dateRange.setEndDate(strEndDate);

ProductSearch productSearch = new ProductSearch();

productSearch.setName(strProductName);

productSearch.setApprovalStatus(productStatus);

productSearch.setSaleDate(dateRange);

SearchProductsRequest searchProductsRequest = new SearchProductsRequest();

searchProductsRequest.setPagingData(requestPagingData);

searchProductsRequest.setAuth(authentication);

searchProductsRequest.setProductSearch(productSearch);

ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();

SearchProductsResponse searchProductsResponse = port.searchProducts(searchProductsRequest);

List<ProductBasic> productList = searchProductsResponse.getProducts().getProduct();

System.out.println("Founded product(s):");

for (ProductBasic sampleProduct:productList

) {

System.out.println(sampleProduct.getId() + " " + sampleProduct.getTitle());

}

}

### N11 Ürün ID’si ile Ürün Silme (DeleteProductById)

Kayıtlı olan bir ürünü N11 Id si kullanarak silmek için kullanılır.

**DeleteProductBuIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Ürün N11 ID bilgisi* |

**\*Bold alanlar zorunlu alanlardır.**

**DeleteProductByIdResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*Bkz:: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**DeleteProductById Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:DeleteProductByIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productId>126473177</productId>

</sch:DeleteProductByIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:DeleteProductByIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>50.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>50.00</displayPrice>

<id>126473177</id>

<price>50.00</price>

<productSellerCode>itemdenemekod1</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>60.00</currencyAmount>

<displayPrice>60.00</displayPrice>

<optionPrice>60.00</optionPrice>

<sellerStockCode>itemdenemestokkod1</sellerStockCode>

<attributes>

<attribute>

<id>354079908</id>

<name>Renk</name>

<value>Sarı</value>

</attribute>

</attributes>

<id>123498755152</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Lorem ipsum dolor sit ametekkod</subtitle>

<title>Lorem ipsumekkod</title>

</product>

</ns3:DeleteProductByIdResponse>

</env:Body>

</env:Envelope>

**DeleteProductById Örnek Çağrı (JAVA)**

public static void main (String [] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

long productID = 126473178;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

DeleteProductByIdRequest deleteProductByIdRequest = new DeleteProductByIdRequest();

deleteProductByIdRequest.setAuth(authentication);

deleteProductByIdRequest.setProductId(productID);

ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();

DeleteProductByIdResponse deleteProductByIdResponse = port.deleteProductById(deleteProductByIdRequest); }

### Mağaza Ürün Kodu ile Ürün Silme(DeleteProductBySellerCode)

Kayıtlı olan bir ürünü mağaza ürün kodu kullanılarak silmek için kullanılır.

**DeleteProductBySellerCodeRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Ürünün mağaza tarafındaki Ürün kodu* |

**\*Bold alanlar zorunlu alanlardır.**

**DeleteProductBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*Bkz:: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**DeleteProductBySellerCode Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:DeleteProductBySellerCodeRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productSellerCode>itemdenemekod1</productSellerCode>

</sch:DeleteProductBySellerCodeRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:DeleteProductBySellerCodeResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>50.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>50.00</displayPrice>

<id>126473179</id>

<price>50.00</price>

<productSellerCode>itemdenemekod1</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>60.00</currencyAmount>

<displayPrice>60.00</displayPrice>

<optionPrice>60.00</optionPrice>

<sellerStockCode>itemdenemestokkod1</sellerStockCode>

<attributes>

<attribute>

<id>354079908</id>

<name>Renk</name>

<value>Sarı</value>

</attribute>

</attributes>

<id>123498755154</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Lorem ipsum dolor sit ametekkod</subtitle>

<title>Lorem ipsumekkod</title>

</product>

</ns3:DeleteProductBySellerCodeResponse>

</env:Body>

</env:Envelope>

**DeleteProductBySellerCode Örnek Çağrı (JAVA)**

public static void main (String [] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String productSellerCode = "itemdenemekod1";

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

DeleteProductBySellerCodeRequest deleteProductBySellerCodeRequest = new DeleteProductBySellerCodeRequest();

deleteProductBySellerCodeRequest.setAuth(authentication);

deleteProductBySellerCodeRequest.setProductSellerCode(productSellerCode);

ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();

DeleteProductBySellerCodeResponse deleteProductBySellerCodeResponse =port.deleteProductBySellerCode(deleteProductBySellerCodeRequest);

ProductBasic deletedProduct = deleteProductBySellerCodeResponse.getProduct();

System.out.println(deletedProduct.getId() + " " + deletedProduct.getTitle() + " has been deleted.");

}

### N11 Ürün ID’si ile İndirim Tutarı Güncelleme (UpdateDiscountValueByProductId)

Bir ürünün N11 ürün ID sini kullanarak indirim bilgilerinin güncellenmesi için kullanılır. Girilen indirim miktarı ürün listeleme fiyatına uygulanır. Liste fiyatı ile ürünün indirimli fiyatı arasındaki fark kadar ürün stok birim fiyatlarına da indirim uygulanır.

**UpdateDiscountValueByProductIdRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Ürünün N11 ürün kodu* |
| ***discountType*** | [*Bkz:: DiscountType*](#_İndirim_Tipi_(Discount) |
| ***productDiscount.discountValue*** | *İndirim miktarı (indirim tipinde verilen parametreye göre)* |
| *productDiscount.discountStartDate* | *Mağaza indirimi başlama tarihi* |
| *productDiscount.discountEndDate* | *Mağaza indirimi bitiş tarihi* |

**\*Bold alanlar zorunlu alanlardır.**

**UpdateDiscountValueByProductIdResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*Bkz:: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**UpdateDiscountValueByProductId Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:UpdateDiscountValueByProductIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productId>126473181</productId>

<productDiscount>

<discountType>2</discountType>

<discountValue>15</discountValue>

<discountStartDate>?</discountStartDate>

<discountEndDate>?</discountEndDate>

</productDiscount>

</sch:UpdateDiscountValueByProductIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:UpdateDiscountValueByProductIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>50.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>42.50</displayPrice>

<id>126473181</id>

<price>50.00</price>

<productSellerCode>itemdenemekod1</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>60.00</currencyAmount>

<displayPrice>51.00</displayPrice>

<optionPrice>60.00</optionPrice>

<sellerStockCode>itemdenemestokkod1</sellerStockCode>

<attributes>

<attribute>

<id>354079908</id>

<name>Renk</name>

<value>Sarı</value>

</attribute>

</attributes>

<id>123498755156</id>

<quantity>5</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Lorem ipsum dolor sit ametekkod</subtitle>

<title>Lorem ipsumekkod</title>

</product>

</ns3:UpdateDiscountValueByProductIdResponse>

</env:Body>

</env:Envelope>

**UpdateDiscountValueByProductId Örnek Çağrı (JAVA)**

public static void main (String [] argv ) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strDiscStartDate = "06.01.2016";

String strDiscEndDate = "24.01.2023";

long discountType = 2;

double discountValue = 15.00;

long productID = 126473181;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

SellerProductDiscount sellerProductDiscount = new SellerProductDiscount();

sellerProductDiscount.setDiscountStartDate(strDiscStartDate);

sellerProductDiscount.setDiscountEndDate(strDiscEndDate);

sellerProductDiscount.setDiscountType(BigInteger.valueOf(discountType));

sellerProductDiscount.setDiscountValue(discountValue);

UpdateDiscountValueByProductIdRequest updateDiscountValueByProductIdRequest = new UpdateDiscountValueByProductIdRequest();

updateDiscountValueByProductIdRequest.setProductId(productID);

updateDiscountValueByProductIdRequest.setProductDiscount(sellerProductDiscount);

updateDiscountValueByProductIdRequest.setAuth(authentication);

ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();

UpdateDiscountValueByProductIdResponse response = port.updateDiscountValueByProductId(updateDiscountValueByProductIdRequest);

ProductBasic discountProduct = response.getProduct();

System.out.println(discountProduct.getId() + " " + discountProduct.getTitle() + " now has a new discount value. Old price: " + discountProduct.getPrice() + ", New Price : " + discountProduct.getDisplayPrice());

**}**

### Mağaza Ürün Kodu ile İndirim Tutarı Güncelleme (UpdateDiscountValueBySellerCode)

Bir ürünün mağaza ürün kodunu kullanarak indirim bilgilerinin güncellenmesi için kullanılır. Girilen indirim miktarı ürün listeleme fiyatına uygulanır. Liste fiyatı ile ürünün indirimli fiyatı arasındaki fark kadar ürün stok birim fiyatlarına da indirim uygulanır.

**UpdateDiscountValueBySellerCodeRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Mağaza ürün kodu* |
| ***discountType*** | [*Bkz:: DiscountType*](#_İndirim_Tipi_(Discount) |
| ***productDiscount.discountValue*** | *İndirim miktarı (indirim tipinde verilen parametreye göre)* |
| *productDiscount.discountStartDate* | *Mağaza indirimi başlama tarihi* |
| *productDiscount.discountEndDate* | *Mağaza indirimi bitiş tarihi* |

**\*Bold alanlar zorunlu alanlardır.**

**UpdateDiscountValueBySellerCode Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:UpdateDiscountValueBySellerCodeRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productSellerCode>4084500051065\_2</productSellerCode>

<productDiscount>

<discountType>1</discountType>

<discountValue>5.00</discountValue>

<discountStartDate>06.01.2016</discountStartDate>

<discountEndDate>24.01.2023</discountEndDate>

</productDiscount>

</sch:UpdateDiscountValueBySellerCodeRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:UpdateDiscountValueBySellerCodeResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>47.95</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>42.95</displayPrice>

<id>119936770</id>

<price>47.95</price>

<productSellerCode>4084500051065\_2</productSellerCode>

<approvalStatus>2</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>47.95</currencyAmount>

<displayPrice>42.95</displayPrice>

<optionPrice>47.95</optionPrice>

<attributes/>

<id>123338487602</id>

<quantity>19</quantity>

<version>2</version>

</stockItem>

</stockItems>

<subtitle>İkili Fırsat Paketi</subtitle>

<title>Head &amp; Shoulders Şampuan Komple Bakım 600 ml</title>

</product>

</ns3:UpdateDiscountValueBySellerCodeResponse>

</env:Body>

</env:Envelope>

**UpdateDiscountValueBySellerCode Örnek Çağrı (JAVA)**

public static void main (String [] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strDiscStartDate = "06.01.2016";

String strDiscEndDate = "24.01.2023";

String strProductSellerCode = "4084500051065\_2";

long discountTypeValue = 1;

double discountValue = 6.00;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

SellerProductDiscount sellerProductDiscount = new SellerProductDiscount();

sellerProductDiscount.setDiscountStartDate(strDiscStartDate);

sellerProductDiscount.setDiscountEndDate(strDiscEndDate);

sellerProductDiscount.setDiscountType(BigInteger.valueOf(discountTypeValue));

sellerProductDiscount.setDiscountValue(discountValue);

UpdateDiscountValueBySellerCodeRequest updateDiscountValueBySellerCodeRequest = new UpdateDiscountValueBySellerCodeRequest();

updateDiscountValueBySellerCodeRequest.setProductSellerCode(strProductSellerCode);

updateDiscountValueBySellerCodeRequest.setProductDiscount(sellerProductDiscount);

updateDiscountValueBySellerCodeRequest.setAuth(authentication);

ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();

UpdateDiscountValueBySellerCodeResponse response = port.updateDiscountValueBySellerCode(updateDiscountValueBySellerCodeRequest);

ProductBasic sampleProduct = response.getProduct();

System.out.println("Product's price " + sampleProduct.getPrice() + " is updated as " + sampleProduct.getDisplayPrice()); }

### N11 Ürün Kodu ile Ürün Fiyatı Güncelleme (UpdateProductPriceById)

Bir ürünün N11 ürün ID si kullanılarak ürünün sadece baz fiyat bilgilerini, ürün stok birimi fiyat bilgilerini veya her ikisinin güncellenmesi için kullanılır.

**UpdateProductPriceByIdRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Ürünün N11 ID si* |
| *price* | *Güncellenmek istenen ürün fiyatı* |
| *currencyType* | *Ürün listeleme fiyatı para birimi* |
| *product.stockItems.stockItem.sellerStockCode* | *Mağaza ürün stok kodu* |
| *product.stockItems.stockItem.optionPrice* | *Ürün stok biriminin liste fiyatı* |

**\*Bold alanlar zorunlu alanlardır.**

**UpdateProductPriceByIdResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*Bkz: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**UpdateProductPriceById Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:UpdateProductPriceByIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productId>127850915</productId>

<price>3.00</price>

<currencyType>TL</currencyType>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<sellerStockCode>stockcode123456</sellerStockCode>

<optionPrice>3</optionPrice>

</stockItem>

</stockItems>

</sch:UpdateProductPriceByIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:UpdateProductPriceByIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>3.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>3.00</displayPrice>

<id>127850915</id>

<oldPrice>5</oldPrice>

<price>3.00</price>

<productSellerCode>12122122223312</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>3.00</currencyAmount>

<displayPrice>3.00</displayPrice>

<oldOptionPrice>5.00</oldOptionPrice>

<optionPrice>3.00</optionPrice>

<sellerStockCode>stockcode123456</sellerStockCode>

<attributes/>

<id>123524100151</id>

<quantity>123123</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Deneme</subtitle>

<title>1Deneme</title>

</product>

</ns3:UpdateProductPriceByIdResponse>

</env:Body>

</env:Envelope>

***N11 Ürün Kodu ile Ürün Fiyatı Güncelleme Örnek Çağrı (JAVA)***

public static void main (String [] args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 String strSellerStockCode = "stockcode123456";  
 long productIdValue = 127850915;  
 int currencyTypeValue = 1;  
 double optionalPriceValue = 5.00;  
 double priceValue = 5.00;  
  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 ProductSkuBasicRequestItemList productSkuBasicRequestItemList = new ProductSkuBasicRequestItemList();  
 ProductSkuRequest productSkuRequest = new ProductSkuRequest();  
 productSkuRequest.setSellerStockCode(strSellerStockCode);  
 productSkuRequest.setOptionPrice(BigDecimal.valueOf(optionalPriceValue));  
  
 UpdateProductPriceByIdRequest updateProductPriceByIdRequest = new UpdateProductPriceByIdRequest();  
 updateProductPriceByIdRequest.setAuth(authentication);  
 updateProductPriceByIdRequest.setProductId(productIdValue);  
 updateProductPriceByIdRequest.setCurrencyType(BigInteger.valueOf(currencyTypeValue));  
 updateProductPriceByIdRequest.setStockItems(productSkuBasicRequestItemList);  
 updateProductPriceByIdRequest.setPrice(BigDecimal.valueOf(priceValue));  
 ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();  
 UpdateProductPriceByIdResponse response = port.updateProductPriceById(updateProductPriceByIdRequest);  
 ProductBasic sampleProduct = response.getProduct();  
  
 System.out.println("Product's price " + sampleProduct.getPrice() + " is updated as " + sampleProduct.getDisplayPrice());  
}

### Mağaza Ürün Kodu ile Ürün Fiyatı Güncelleme (UpdateProductPriceBySellerCode)

Bir ürünün mağaza ürün kodu kullanarak fiyat bilgilerinin güncellenmesi için kullanılır.

**UpdateProductPriceBySellerCodeRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Ürünün mağaza tarafındaki ürün kodu* |
| ***price*** | *Güncellenmek istenen ürün baz fiyatı* |
| *currencyType* | *Ürün listeleme fiyatı para birimi* |
| *product.stockItems.stockItem.sellerStockCode* | *Mağaza ürün stok kodu* |
| *product.stockItems.stockItem.optionPrice* | *Ürün stok biriminin liste fiyatı* |

**\*Bold alanlar zorunlu alanlardır.**

**UpdateProductPriceBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*Bkz:: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**UpdateProductPriceBySellerCode Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:UpdateProductPriceBySellerCodeRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productSellerCode>12122122223312</productSellerCode>

<price>22.00</price>

<currencyType>1</currencyType>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<sellerStockCode>stockcode123456</sellerStockCode>

<optionPrice>22.00</optionPrice>

</stockItem>

</stockItems>

</sch:UpdateProductPriceBySellerCodeRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:UpdateProductPriceBySellerCodeResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>22.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>22.00</displayPrice>

<id>127850915</id>

<oldPrice>5</oldPrice>

<price>22.00</price>

<productSellerCode>12122122223312</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>22.00</currencyAmount>

<displayPrice>22.00</displayPrice>

<oldOptionPrice>5.00</oldOptionPrice>

<optionPrice>22.00</optionPrice>

<sellerStockCode>stockcode123456</sellerStockCode>

<attributes/>

<id>123524100151</id>

<quantity>123123</quantity>

<version>0</version>

</stockItem>

</stockItems>

<subtitle>Deneme</subtitle>

<title>1Deneme</title>

</product>

</ns3:UpdateProductPriceBySellerCodeResponse>

</env:Body>

</env:Envelope>

**UpdateProductPriceBySellerCode Örnek Çağrı (JAVA)**

public static void main (String [] args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 String strSellerCode = "12122122223312";  
 String strSellerStockCode = "stockcode123456";  
 int currencyTypeValue = 1;  
 double optionalPriceValue = 5.00;  
 double priceValue = 5.00;  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 ProductSkuBasicRequestItemList productSkuBasicRequestItemList = new ProductSkuBasicRequestItemList();  
 ProductSkuRequest productSkuRequest = new ProductSkuRequest();  
 productSkuRequest.setSellerStockCode(strSellerStockCode);  
 productSkuRequest.setOptionPrice(BigDecimal.valueOf(optionalPriceValue));  
  
 UpdateProductPriceBySellerCodeRequest request = new UpdateProductPriceBySellerCodeRequest();  
 request.setAuth(authentication);  
 request.setCurrencyType(BigInteger.valueOf(currencyTypeValue));  
 request.setStockItems(productSkuBasicRequestItemList);  
 request.setPrice(BigDecimal.valueOf(priceValue));  
 request.setProductSellerCode(strSellerCode);  
  
 ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();  
 UpdateProductPriceBySellerCodeResponse response = port.updateProductPriceBySellerCode(request);  
 ProductBasic sampleProduct = response.getProduct();

}

### Temel Ürün Özellikleri Güncelleme (UpdateProductBasic)

Kayıtlı olan bir ürünün, N11 ürün ID’si ya da mağaza kodu ile ürün fiyatını, ürün üzerindeki indirimi ve isteğe bağlı olarak stok ID’si ya da mağaza stok kodu ile belirtilen stoklarının, miktarı ve ilgili stok biriminin liste fiyatının güncellenmesi için kullanılır.

**UpdateProductBasicRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| *productId* | *Ürün N11 ID bilgisi* |
| *productSellerCode* | *Ürün mağaza kodu* |
| *price* | *Ürünün baz fiyatı* |
| *productDiscount.discountType* | [*Bkz:: DiscountType*](#_İndirim_Tipi_(Discount) |
| *productDiscount.discountValue* | *İndirim Miktarı (indirim tipinde verilen parametreye göre)* |
| *productDiscount.discountStartDate* | *Mağaza indirimi başlama tarihi* |
| *productDiscount.discountEndDate* | *Mağaza indirimi bitiş tarihi* |
| *stockItems.stockItem.sellerStockCode* | *Ürün stok mağaza kodu* |
| *stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *stockItems.stockItem.optionPrice* | *Ürün stok biriminin liste fiyatı* |
| *stockItems.stockItem.quantity* | *Ürün stok miktarı* |

**\*Bold alanlar zorunlu alanlardır.**

**UpdateProductBasicResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*Bkz: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**UpdateProductBasic Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:UpdateProductBasicRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productId>258491025</productId>

<productSellerCode/>

<price>125</price>

<productDiscount>

<discountType>2</discountType>

<discountValue>25</discountValue>

<discountStartDate/>

<discountEndDate/>

</productDiscount>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<sellerStockCode></sellerStockCode>

<id></id>

<optionPrice></optionPrice>

<quantity></quantity>

</stockItem>

</stockItems>

</sch:UpdateProductBasicRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:UpdateProductBasicResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>125.00</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>93.75</displayPrice>

<id>258491025</id>

<oldPrice>1232</oldPrice>

<price>125.00</price>

<productSellerCode>2</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<bundle>false</bundle>

<currencyAmount>125.00</currencyAmount>

<displayPrice>93.75</displayPrice>

<oldOptionPrice>1232.00</oldOptionPrice>

<optionPrice>125.00</optionPrice>

<attributes/>

<id>122375265401</id>

<quantity>3</quantity>

<version>8</version>

</stockItem>

</stockItems>

<subtitle>Bilgisayar</subtitle>

<title>Dizüstü Bilgisayar</title>

</product>

</ns3:UpdateProductBasicResponse>

</env:Body>

</env:Envelope>

**GetProductQuestionList Örnek Çağrı (JAVA)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strProdSellerCode = "";

String strDiscountStartDate = "";

String strDiscountEndDate = "";

String strSellerStockCode = "";

double priceDec = 125.00;

long productIdVal = 258491025;

int discountTypeVal = 2;

int discountValueVal = 25;

int stockQuantityVal = 0;

int stockIdVal = 0;

double stockOptionPriceVal = 0;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

SellerProductDiscount sellerProductDiscount = new SellerProductDiscount();

sellerProductDiscount.setDiscountType(BigInteger.valueOf(discountTypeVal));

sellerProductDiscount.setDiscountValue(discountValueVal);

sellerProductDiscount.setDiscountStartDate(strDiscountStartDate);

sellerProductDiscount.setDiscountEndDate(strDiscountEndDate);

ProductUpdateSkuBasicRequest productUpdateSkuBasicRequest = new ProductUpdateSkuBasicRequest();

productUpdateSkuBasicRequest.setSellerStockCode(strSellerStockCode);

productUpdateSkuBasicRequest.setOptionPrice(BigDecimal.valueOf(stockOptionPriceVal));

productUpdateSkuBasicRequest.setId(stockIdVal);

ProductUpdateSkuBasicRequestItemList productUpdateSkuBasicRequestItemList

= new ProductUpdateSkuBasicRequestItemList();

productUpdateSkuBasicRequestItemList.getStockItem().add(productUpdateSkuBasicRequest);

UpdateProductBasicRequest request = new UpdateProductBasicRequest();

request.setProductSellerCode(strProdSellerCode);

request.setPrice(BigDecimal.valueOf(priceDec));

request.setAuth(authentication);

request.setProductId(productIdVal);

request.setProductDiscount(sellerProductDiscount);

request.setStockItems(productUpdateSkuBasicRequestItemList);

ProductServicePort port =

new ProductServicePortService().getProductServicePortSoap11();

UpdateProductBasicResponse response = port.updateProductBasic(request);

System.out.println(response.getProduct().getId() + " Display price:"

+ response.getProduct().getDisplayPrice() + " Price:" + response.getProduct().getPrice());

}

### Ürün Sorularını Listeleme (GetProductQuestionList)

*Müşterileriniz tarafından mağazanıza sorulan soruları listeler. Sorularınızı listelemek için Appkey ve Appsecret bilgileriniz gerekmektedir.*

**GetProductQuestionListRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| *productQuestionSearch.productId* | *Ürün N11 ID si* |
| *productQuestionSearch.buyerEmail* | *Müşteri e-posta adresi* |
| *productQuestionSearch.subject* | *Sorunun konusu* |
| *productQuestionSearch.status* | *Gelen soruların durumları* |
| *productQuestionSearch.questionDate* | *Sorunun gelme tarihi* |
| ***currentPage*** | *Gösterilen sayfa numarası (currentpage 0 ile başlar)* |
| ***pageSize*** | *Sayfada yer alan madde sayısı* |

**\*Bold alanlar zorunlu alanlardır.**

**GetProductQuestionListResponse**

|  |  |
| --- | --- |
| *ProductQuestionId* | *Gelen sorunun N11 tarafındaki soru ID si.* |
| *productQuestion.productId* | *Ürün N11 tarafında sahip olduğu ID* |
| *productQuestion.producTitle* | *Ürün başlığı* |
| *productQuestion.question* | *Gelen sorunun açıklaması* |
| *productQuestion.subject* | *Sorunun konusu* |
| *productQuestion.questionDate* | *Sorunun gelme tarihi* |
| *currentPage* | *Gösterilen sayfa numarası (currentpage 0 ile başlar)* |
| *pageSize* | *Sayfada yer alan madde sayısı* |
| *Result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |

**GetProductQuestionList Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetProductQuestionListRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productQuestionSearch>

<productId>122879934</productId>

<buyerEmail></buyerEmail>

<subject></subject>

<status></status>

<questionDate>07/09/2016</questionDate>

</productQuestionSearch>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

</pagingData>

</sch:GetProductQuestionListRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetProductQuestionListResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<productQuestions>

<productQuestion>

<id>13479715</id>

<productId>122879934</productId>

<productTitle>Ariel Dağ Esintisi 9 Kg + Parlak Renkler Çamaşır Deterjanı 7 Kg</productTitle>

<question>Eylülün 2 sinde sipariş verdim hala kargolanmamış</question>

<questionSubject>Ariel deterjan</questionSubject>

</productQuestion>

</productQuestions>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

<totalCount>1</totalCount>

<pageCount>1</pageCount>

</pagingData>

</ns3:GetProductQuestionListResponse>

</env:Body>

</env:Envelope>

**GetProductQuestionList Örnek Çağrı (JAVA)**

public static void main (String []args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 String strQuestionDate = "07/09/2016";  
 long productIdValue = 122879934;  
 int currentPageValue = 0;  
 int pageSizeValue = 3;  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 RequestPagingData requestPagingData = new RequestPagingData();  
 requestPagingData.setCurrentPage(currentPageValue);  
 requestPagingData.setPageSize(pageSizeValue);  
  
 ProductQuestionSearch productQuestionSearch = new ProductQuestionSearch();  
 productQuestionSearch.setProductId(productIdValue);  
 productQuestionSearch.setQuestionDate(strQuestionDate);  
  
 GetProductQuestionListRequest request = new GetProductQuestionListRequest();  
 request.setAuth(authentication);  
 request.setPagingData(requestPagingData);  
 request.setProductQuestionSearch(productQuestionSearch);  
  
 ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();  
 GetProductQuestionListResponse response = port.getProductQuestionList(request);  
 List<ProductQuestion> productQuestionList = response.getProductQuestions().getProductQuestion();  
  
 for (ProductQuestion sampleProductQuestion:productQuestionList  
 ) {  
 System.out.println("Subject:" + sampleProductQuestion.getQuestionSubject() + " Question: " + sampleProductQuestion.getQuestion());  
 }  
}

### Ürün Sorularını Detaylandırma (GetProductQuestionDetail)

*GetProductQuestionList ile sıralanan soruların içeriğini, buradan gelen ID ve getProductQuestionDetail yardımıyla görüntüleyebilirsiniz.*

**GetProductQuestionDetailRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productQuestionId*** | *Gelen Sorunun N11 id’si* |

**\*Bold alanlar zorunlu alanlardır.**

**GetProductQuestionDetailResponse**

|  |  |
| --- | --- |
| *Result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *productQuestion.answeredDate* | *Cevaplanma tarihi* |
| *productQuestion.buyerExpose* | *Gelen sorunun durumu (özel, genel gibi)* |
| *productQuestion.Email* | *Müşteri e-posta adresi* |
| *productQuestion.FullName* | *Müşteri adı* |
| *productQuestion.productId* | *Ürün ID* |
| *productQuestion.productStatus* | *Ürün durumu* |
| *productQuestion.productTitle* | *Ürün başlığı* |
| *productQuestion.question* | *Soru açıklaması* |
| *productQuestion.questionDate* | *Soru tarihi* |
| *productQuestion.questionSubject* | *Sorunun konusu* |
| *productQuestion.sellerExpose* | *Cevaplanan sorunun durumu (özel,genel gibi)* |
| *Status* | *Sorunun durumu* |

**GetProductQuestionDetail Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetProductQuestionDetailRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productQuestionId>13479715</productQuestionId>

</sch:GetProductQuestionDetailRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"> <env:Header/>

<env:Body>

<ns3:GetProductQuestionDetailResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<productQuestion>

<buyerExpose>NO</buyerExpose>

<email>n11@pismail.com</email>

<fullName>\*\*\*\*\*\*\*\*\*\*\*\*</fullName>

<productId>122879934</productId>

<productStatus>Active</productStatus>

<productTitle>Ariel Dağ Esintisi 9 Kg + Parlak Renkler Çamaşır Deterjanı 7 Kg</productTitle>

<question>Eylülün 2 sinde sipariş verdim hala kargolanmamış</question>

<questionDate>07/09/2016</questionDate>

<questionSubject>Ariel deterjan</questionSubject>

<sellerExpose>YES</sellerExpose>

<status>OPEN</status>

</productQuestion>

</ns3:GetProductQuestionDetailResponse>

</env:Body>

</env:Envelope>

**GetProductQuestionDetail Örnek Çağrı (JAVA)**

public static void main (String []args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 long productQuestionIdValue = 13479715;  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 GetProductQuestionDetailRequest request = new GetProductQuestionDetailRequest();  
 request.setProductQuestionId(productQuestionIdValue);  
 request.setAuth(authentication);  
  
 ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();  
 GetProductQuestionDetailResponse response = port.getProductQuestionDetail(request);  
 ProductQuestionDetail productQuestionDetail = response.getProductQuestion();  
  
 System.out.println(productQuestionDetail.getEmail() + " " + productQuestionDetail.getQuestion());  
}

### Ürün Sorularını Cevaplandırma (SaveProductAnswer)

*Müşterilerden gelen ürün sorularını cevaplamak için kullanılır. Cevap vermek için* ***productQuestionId*** *değeri zorunludur ve GetProductQuestionList‘ten id edinilebilir.*

**SaveProductAnswerRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productQuestionId*** | *Gelen Sorunun N11 ID’si* |
| ***productAnswer*** | *Sorunun cevabı* |

**\*Bold alanlar zorunlu alanlardır.**

**SaveProductAnswerResponse**

|  |  |
| --- | --- |
| ***result*** | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |

**SaveProductAnswer Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetProductQuestionDetailRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productQuestionId>13479715</productQuestionId>

</sch:GetProductQuestionDetailRequest>

</soapenv:Body>

</soapenv:Envelope>

***Response***

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:SaveProductAnswerResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

</ns3:SaveProductAnswerResponse>

</env:Body> </env:Envelope>

**SaveProductAnswer Örnek Çağrı (JAVA)**

public static void main (String []args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 long productQuestionIdValue = 13479715;  
 String strAnswer = "This is your answering text.";  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 SaveProductAnswerRequest request = new SaveProductAnswerRequest();  
 request.setProductQuestionId(productQuestionIdValue);  
 request.setAuth(authentication);  
 request.setAnswer(strAnswer);  
  
 ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();  
 SaveProductAnswerResponse response = port.saveProductAnswer(request);  
  
 System.*out*.println(response.getResult().getStatus().getValue());  
}

## 3.4 Ürün Satış Servisleri (ProductSellingService)

Ürün satış servisleri için WSDL adresi:

<https://api.n11.com/ws/ProductSellingService.wsdl>

### N11 Ürün ID’si ile Ürünün Satışını Başlat (StartSellingProductByProductId)

Satışta olmayan bir ürünün N11 ürün ID si kullanılarak satışa başlanması için kullanılır.

**StartSellingProductByProductIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Ürünün n11 tarafındaki ürün ID’si* |

**\*Bold alanlar zorunlu alanlardır.**

**StartSellingProductByProductIdResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*Bkz:: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**StartSellingProductByProductId Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:StartSellingProductByProductIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productId>122879934</productId>

</sch:StartSellingProductByProductIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:StartSellingProductByProductIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>69.90</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>52.90</displayPrice>

<id>122879934</id>

<price>69.90</price>

<productSellerCode>8681002941960</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>69.90</currencyAmount>

<displayPrice>52.90</displayPrice>

<optionPrice>69.90</optionPrice>

<attributes/>

<id>123423877639</id>

<quantity>130</quantity>

<version>1411</version>

</stockItem>

</stockItems>

<subtitle>İkili Fırsat Paket ( Ücretsiz Kargo )</subtitle>

<title>Ariel Dağ Esintisi 9 Kg + Parlak Renkler Çamaşır Deterjanı 7 Kg</title>

</product>

</ns3:StartSellingProductByProductIdResponse>

</env:Body>

</env:Envelope>

**StartSellingProductByProductId Örnek Çağrı (JAVA)**

public static void main(String[] args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 long productIdValue = 16589849;  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 StartSellingProductByProductIdRequest request = new StartSellingProductByProductIdRequest();  
 request.setAuth(authentication);  
 request.setProductId(productIdValue);  
 ProductSellingServicePort port = new ProductSellingServicePortService().getProductSellingServicePortSoap11();  
 StartSellingProductByProductIdResponse response = port.startSellingProductByProductId(request);  
 ProductBasic sampleProduct = response.getProduct();  
  
 System.out.println(sampleProduct.getId() + " has been activated.");  
}

### Mağaza Ürün Kodu ile Ürünün Satışını Başlat (StartSellingProductBySellerCode)

Satışta olmayan bir ürünün mağaza ürün kodu kullanılarak satışa başlanması için kullanılır.

**StartSellingProductByProductIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Ürünün mağaza tarafındaki ürün kodu* |

**\*Bold alanlar zorunlu alanlardır.**

**StartSellingProductBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*Bkz:: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**StartSellingProductBySellerCode Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:StartSellingProductBySellerCodeRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productSellerCode>8681002941960</productSellerCode>

</sch:StartSellingProductBySellerCodeRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:StartSellingProductBySellerCodeResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>69.90</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>52.90</displayPrice>

<id>122879934</id>

<price>69.90</price>

<productSellerCode>8681002941960</productSellerCode>

<approvalStatus>1</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>69.90</currencyAmount>

<displayPrice>52.90</displayPrice>

<optionPrice>69.90</optionPrice>

<attributes/>

<id>123423877639</id>

<quantity>130</quantity>

<version>1411</version>

</stockItem>

</stockItems>

<subtitle>İkili Fırsat Paket ( Ücretsiz Kargo )</subtitle>

<title>Ariel Dağ Esintisi 9 Kg + Parlak Renkler Çamaşır Deterjanı 7 Kg</title>

</product>

</ns3:StartSellingProductBySellerCodeResponse>

</env:Body>

</env:Envelope>

**StartSellingProductBySellerCode Örnek Çağrı (JAVA)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strSellerCode = "8681002941960";

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

StartSellingProductBySellerCodeRequest request = new StartSellingProductBySellerCodeRequest();

request.setAuth(authentication);

request.setProductSellerCode(strSellerCode);

ProductSellingServicePort port = new ProductSellingServicePortService().getProductSellingServicePortSoap11();

StartSellingProductBySellerCodeResponse response = port.startSellingProductBySellerCode(request);

ProductBasic sampleProduct = response.getProduct();

System.out.println(sampleProduct.getId() + " has been activated.");

}

### N11 Ürün ID’si ile Ürünün Satışını Durdur (StopSellingProductByProductId)

Satışta olan ürünün n11 ürün ID si kullanılarak satışa kapatılması için kullanılır.

**StopSellingProductByProductIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Ürünün n11 tarafındaki Ürün kodu* |

**\*Bold alanlar zorunlu alanlardır.**

**StopSellingProductByProductIdResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*Bkz: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**StopSellingProductByProductId Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:StopSellingProductByProductIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productId>122879934</productId>

</sch:StopSellingProductByProductIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:StopSellingProductByProductIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

*<result>*

*<status>success</status>*

</result> <product>

<displayPrice>2112.20</displayPrice>

<id>5179</id>

<price>2112.20</price>

<productSellerCode>4022</productSellerCode>

<approvalStatus>2</approvalStatus>

<saleStatus>2</saleStatus>

<subtitle>ürün altbaşlığı</subtitle>

<title>Ürün başlığı</title>

</product>

</ns3:StopSellingProductByProductIdResponse>

</env:Body>

</env:Envelope>

**StopSellingProductByProductId Örnek Çağrı (JAVA)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

long productIdValue = 122879934;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

StopSellingProductByProductIdRequest request = new StopSellingProductByProductIdRequest();

request.setAuth(authentication);

request.setProductId(productIdValue);

ProductSellingServicePort port = new ProductSellingServicePortService().getProductSellingServicePortSoap11();

StopSellingProductByProductIdResponse response = port.stopSellingProductByProductId(request);

ProductBasic sampleProduct = response.getProduct();

System.out.println(sampleProduct.getId() + " has been stopped.");

}

### Mağaza Ürün Kodu ile Ürünün Satışını Durdur (StopSellingProductBySellerCode)

Satışta olan ürünün mağaza ürün kodu kullanılarak satışının durdurulması için kullanılır.

**StopSellingProductBySellerCodeRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Ürünün mağaza tarafındaki Ürün kodu* |

**\*Bold alanlar zorunlu alanlardır.**

**StopSellingProductBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*Bkz: Ürün Özet Bilgisi*](#_Ürün_Özet_Bilgi) |

**StopSellingProductBySellerCode Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:StopSellingProductBySellerCodeRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productSellerCode>8681002941960</productSellerCode>

</sch:StopSellingProductBySellerCodeRequest>

</soapenv:Body> </soapenv:Envelope>

***Response***

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:StopSellingProductBySellerCodeResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<product>

<currencyAmount>69.90</currencyAmount>

<currencyType>1</currencyType>

<displayPrice>52.90</displayPrice>

<id>122879934</id>

<price>69.90</price>

<productSellerCode>8681002941960</productSellerCode>

<approvalStatus>2</approvalStatus>

<saleStatus>2</saleStatus>

<stockItems>

<stockItem>

<currencyAmount>69.90</currencyAmount>

<displayPrice>52.90</displayPrice>

<optionPrice>69.90</optionPrice>

<attributes/>

<id>123423877639</id>

<quantity>130</quantity>

<version>1411</version>

</stockItem>

</stockItems>

<subtitle>İkili Fırsat Paket ( Ücretsiz Kargo )</subtitle>

<title>Ariel Dağ Esintisi 9 Kg + Parlak Renkler Çamaşır Deterjanı 7 Kg</title>

</product>

</ns3:StopSellingProductBySellerCodeResponse>

</env:Body>

</env:Envelope>

**StopSellingProductBySellerCode Örnek Çağrı (JAVA)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strSellerCode = "8681002941960";

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

StopSellingProductBySellerCodeRequest request = new StopSellingProductBySellerCodeRequest();

request.setAuth(authentication);

request.setProductSellerCode(strSellerCode);

ProductSellingServicePort port = new ProductSellingServicePortService().getProductSellingServicePortSoap11();

StopSellingProductBySellerCodeResponse response = port.stopSellingProductBySellerCode(request);

ProductBasic sampleProduct = response.getProduct();

System.out.println(sampleProduct.getId() + " has been stopped.");

}

## 3.5 Ürün Stok Servisleri (ProductStockService)

Ürün stok servisleri için WSDL adresi:

<https://api.n11.com/ws/ProductStockService.wsdl>

### N11 Ürün ID si ile Stok Bilgisi Listeleme (GetProductStockByProductId)

Sistemde kayıtlı olan ürünün N11 ürün ID si ile ürün stok bilgilerini getiren metottur.

Cevap içinde stok durumunun “**version**” bilgisi de vardır, ürün stoklarında değişme olduysa bu versiyon bilgisi artacaktır, çağrı yapan taraf versiyon bilgisini kontrol ederek N11 e verilen stok bilgilerinde değişim olup olmadığını anlayabilir.

***GetProductStockByProductIdRequest***

|  |  |
| --- | --- |
| ***auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***product.Id*** | *Ürün ID bilgisi* |

**\*Bold alanlar zorunlu alanlardır.**

***GetProductStockByProductIdResponse***

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.id* | *Stok tanımının olduğu seçenek idsi* |
| *stockItems.stockItem.attributes.atribute.name* | *Stok tanımının olduğu seçenek adı* |
| *stockItems.stockItem.attributes.atribute.value* | *Stok tanımının olduğu seçenek değeri* |
| *stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *stockItems.stockItem.sellerCode* | *Ürün stok mağaza kodu* |
| *stockItems.stockItem.quantity* | *Ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**GetProductStockByProductId Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetProductStockByProductIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productId>122879934</productId>

</sch:GetProductStockByProductIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetProductStockByProductIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<stockItems>

<stockItem>

<currencyAmount>69.90</currencyAmount>

<displayPrice>52.90</displayPrice>

<optionPrice>69.90</optionPrice>

<attributes/>

<id>123423877639</id>

<quantity>130</quantity>

<version>1411</version>

</stockItem>

</stockItems>

</ns3:GetProductStockByProductIdResponse>

</env:Body>

</env:Envelope>

**GetProductStockByProductId Örnek Çağrı (JAVA)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

long productIdValue = 122879934;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

GetProductStockByProductIdRequest request = new GetProductStockByProductIdRequest();

request.setAuth(authentication);

request.setProductId(productIdValue);

ProductStockServicePort port = new ProductStockServicePortService().getProductStockServicePortSoap11();

GetProductStockByProductIdResponse response = port.getProductStockByProductId(request);

List<ProductSku> stockList = response.getStockItems().getStockItem();

for (ProductSku sampleStock:stockList) {

System.out.println(response.getResult().getStatus().getValue() + " Quantity:" + sampleStock.getQuantity());

}

}

### Mağaza Ürün Kodu ile Stok Bilgisi Listeleme (GetProductStockBySellerCode)

Sistemde kayıtlı olan ürünün mağaza ürün kodu ile ürün stok bilgilerini getiren metottur.

Cevap içinde stok durumunun “**version**” bilgisi de vardır, ürün stoklarında değişme olduysa bu versiyon bilgisi artacaktır, çağrı yapan taraf versiyon bilgisini kontrol ederek N11 e verilen stok bilgilerinde değişim olup olmadığını anlayabilir.

***GetProductStockBySellerCodeRequest***

|  |  |
| --- | --- |
| ***auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Mağaza ürün kodu* |

**\*Bold alanlar zorunlu alanlardır.**

***GetProductStockBySellerCodeResponse***

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.id* | *Stok tanımının olduğu seçenek idsi* |
| *stockItems.stockItem.attributes.atribute.name* | *Stok tanımının olduğu seçenek adı* |
| *stockItems.stockItem.attributes.atribute.value* | *Stok tanımının olduğu seçenek değeri* |
| *stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *stockItems.stockItem.sellerCode* | *Ürün stok mağaza kodu* |
| *stockItems.stockItem.quantity* | *Ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**GetProductStockBySellerCode Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetProductStockByProductSellerCodeRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<productSellerCode>8681002941960</productSellerCode>

</sch:GetProductStockByProductSellerCodeRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetProductStockByProductSellerCodeResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<stockItems>

<stockItem>

<currencyAmount>69.90</currencyAmount>

<displayPrice>52.90</displayPrice>

<optionPrice>69.90</optionPrice>

<attributes/>

<id>123423877639</id>

<quantity>130</quantity>

<version>1411</version>

</stockItem>

</stockItems>

</ns3:GetProductStockByProductSellerCodeResponse>

</env:Body>

</env:Envelope>

**GetProductStockBySellerCode Örnek Çağrı (JAVA)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strSellerCode = "8681002941960";

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

GetProductStockByProductSellerCodeRequest request = new GetProductStockByProductSellerCodeRequest();

request.setAuth(authentication);

request.setProductSellerCode(strSellerCode);

ProductStockServicePort port = new ProductStockServicePortService().getProductStockServicePortSoap11();

GetProductStockByProductSellerCodeResponse response = port.getProductStockByProductSellerCode(request);

List<ProductSku> stockList = response.getStockItems().getStockItem();

for (ProductSku sampleStock:stockList) {

System.out.println(response.getResult().getStatus().getValue() + " Quantity:" + sampleStock.getQuantity());

}

}

### Stok Özellikleri ile Stok Miktarı Güncelleme (DeleteAndUpdateStockByStockAttributes )

Ürün stok seçenek bilgilerini kullanarak kayıtlı ürünün stok bilgilerini güncellemek için kullanılır. Ürün n11 ID si ve stok seçenek bilgileri girilerek güncelleme işlemi yapılır. Bir ürüne ait stok bilgilerine, **ProductStockService** içindeki **GetProductStockByProductId** veya **GetProductStockBySellerCode** metotları ile ulaşılabilir. Bir ürün için tüm stok bilgilerini güncelleme işlemi gerçekleştirilebilir.

N11 tarafında değişen stok miktarlarını ezmemek için, “**version**” bilgisi verilmesi durumunda ilgili ürün stok bilgisinin N11 de versiyonu ile karşılaştırma yapılır, stok versiyon numaraları uyumsuz ise işlem gerçekleştirilmez.

***DeleteAndUpdateStockByStockAttributesRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***product.Id*** | *Ürün ID bilgisi* |
| ***product.stockItems.stockItem.attributes.attribute.name*** | *Ürün stok özellik adı* |
| ***product.stockItems.stockItem.attributes.attribute.value*** | *Ürün stok özellik değeri* |
| ***product.stockItems.stockItem.quantity*** | *Güncellenmek istenen ürün stok miktarı* |
| *product.stockItems.stockItem.version* | *Ürün versiyon numarası* |

**\*Bold alanlar zorunlu alanlardır.**

***DeleteAndUpdateStockByStockAttributesResponse***

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Stok tanımının olduğu seçenek adı* |
| *stockItems.stockItem.attributes.atribute.value* | *Stok tanımının olduğu seçenek değeri* |
| *stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *stockItems.stockItem.sellerStockCode* | *Ürün stok mağaza kodu* |
| *stockItems.stockItem.stock* | *Ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**DeleteAndUpdateStockByStockAttributes Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:DeleteAndUpdateStockByStockAttributesRequest>

<auth>

<appKey>b65e6cb4-361f-48ad-9490-284efae40ef4</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<product>

<id>125310829</id>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<attributes>

<!--1 or more repetitions:-->

<attribute>

<name>Beden</name>

<value>3</value>

</attribute>

</attributes>

<quantity>10</quantity>

<version>1</version>

</stockItem>

</stockItems>

</product>

</sch:DeleteAndUpdateStockByStockAttributesRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:DeleteAndUpdateStockByStockAttributesResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<stockItems>

<stockItem>

<attributes>

<attribute>

<name>Beden</name>

<value>3</value>

</attribute>

</attributes>

<id>123568093150</id>

<sellerStockCode>8697472000427</sellerStockCode>

<stock>10</stock>

<version>0</version>

</stockItem>

</stockItems>

</ns3:DeleteAndUpdateStockByStockAttributesResponse>

</env:Body>

</env:Envelope>

**DeleteAndUpdateStockByStockAttributes Örnek Çağrı (JAVA)**

public static void main (String [] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strStoAttributeName = "Beden";

String strStoAttributeValue = "25";

long productIdValue = 125310829;

long versionValue = 4;

int quantityValue = 39;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

StockAttribute attribute = new StockAttribute();

attribute.setName(strStoAttributeName);

attribute.setValue(strStoAttributeValue);

StockAttributeList attributeList = new StockAttributeList();

attributeList.getAttribute().add(attribute);

StockItemForUpdateStockWithAttributes item = new StockItemForUpdateStockWithAttributes();

item.setAttributes(attributeList);

item.setQuantity(BigInteger.valueOf(quantityValue));

item.setVersion(versionValue);

StockItemForUpdateStockWithAttributesList stockItemForUpdateStockWithAttributesList = new StockItemForUpdateStockWithAttributesList();

List<StockItemForUpdateStockWithAttributes> stockList = stockItemForUpdateStockWithAttributesList.getStockItem();

stockList.add(item);

stockItemForUpdateStockWithAttributesList.getStockItem().add(stockList.get(0));

ProductWithIdAndStockAttributesForUpdate productWithIdAndStockAttributesForUpdate = new ProductWithIdAndStockAttributesForUpdate();

productWithIdAndStockAttributesForUpdate.setId(productIdValue);

productWithIdAndStockAttributesForUpdate.setStockItems(stockItemForUpdateStockWithAttributesList);

DeleteAndUpdateStockByStockAttributesRequest request = new DeleteAndUpdateStockByStockAttributesRequest();

request.setAuth(authentication);

request.setProduct(productWithIdAndStockAttributesForUpdate);

ProductStockServicePort port = new ProductStockServicePortService().getProductStockServicePortSoap11();

DeleteAndUpdateStockByStockAttributesResponse response = port.deleteAndUpdateStockByStockAttributes(request);

List<StockItemList> stockItemList = response.getStockItems();

for (StockItemList sampleStockItemList:stockItemList

) {

sampleStockItemList.getStockItem();

}

}

### N11 Ürün Stok ID ile Stok Güncelleme (UpdateStockByStockId)

n11 ürün stok ID si kullanarak kayıtlı ürünün stok bilgilerini güncellemek için kullanılır. n11 ürün stok ID si ve miktar bilgileri girilerek güncelleme işlemi yapılır. Bir ürüne ait n11 ürün stok ID sine, **ProductStockService** içindeki **GetProductStockByProductId** veya **GetProductStockBySellerCode** metotları ile ulaşılabilir.

N11 tarafında değişen stok miktarlarını ezmemek için, “**version**” bilgisi verilmesi durumunda ilgili ürün stok bilgisinin N11 de versiyonu ile karşılaştırma yapılır, stok versiyon numaraları uyumsuz ise işlem gerçekleştirilmez.

***UpdateStockByStockId Request***

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***stockItems.stockItem.id*** | *Ürün stok N11 ID si* |
| ***stockItems.stockItem.quantity*** | *Yeni ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**\*Bold alanlar zorunlu alanlardır.**

***UpdateStockByStockIdResponse***

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Stok tanımının olduğu seçenek adı* |
| *stockItems.stockItem.attributes.atribute.value* | *Stok tanımının olduğu seçenek değeri* |
| *stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *stockItems.stockItem.sellerStockCode* | *Ürün stok mağaza kodu* |
| *stockItems.stockItem.stock* | *Ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**UpdateStockByStockId Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:UpdateStockByStockIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<id>122375331751</id>

<quantity>10</quantity>

<version>1</version>

<delist>1</delist>

</stockItem>

</stockItems>

</sch:UpdateStockByStockIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:UpdateStockByStockIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<stockItems>

<stockItem>

<attributes>

<attribute>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<id>122375331751</id>

<sellerStockCode>StokKod1234567</sellerStockCode>

<stock>10</stock>

<version>1</version>

</stockItem>

</stockItems>

</ns3:UpdateStockByStockIdResponse>

</env:Body>

</env:Envelope>

**UpdateStockByStockId Örnek Çağrı (JAVA)**

public static void main (String [] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

long versionValue = 0;

long stockIdValue = 122375331751L;

int quantityValue = 10;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

StockItemForUpdateStockWithIdList stockItemList = new StockItemForUpdateStockWithIdList();

StockItemForUpdateStockWithId stockItem = new StockItemForUpdateStockWithId();

stockItem.setVersion(versionValue);

stockItem.setId(stockIdValue);

stockItem.setQuantity(BigInteger.valueOf(quantityValue));

stockItemList.getStockItem().add(stockItem);

UpdateStockByStockIdRequest updateStockByStockIdRequest = new UpdateStockByStockIdRequest();

updateStockByStockIdRequest.setAuth(authentication);

updateStockByStockIdRequest.setStockItems(stockItemList);

ProductStockServicePort port = new ProductStockServicePortService().getProductStockServicePortSoap11();

UpdateStockByStockIdResponse response = port.updateStockByStockId(updateStockByStockIdRequest);

System.out.println("Update status is " + response.getResult().getStatus().getValue());

}

### Mağaza Stok Kodu ile Stok Güncelleme (UpdateStockByStockSellerCode)

Mağaza ürün stok kodu kullanarak kayıtlı ürünün stok bilgilerini güncellemek için kullanılır. Mağaza ürün stok kodu ve miktar bilgileri girilerek güncelleme işlemi yapılır.

N11 tarafında değişen stok miktarlarını ezmemek için, “**version**” bilgisi verilmesi durumunda ilgili ürün stok bilgisinin N11 de versiyonu ile karşılaştırma yapılır, stok versiyon numaraları uyumsuz ise işlem gerçekleştirilmez.

***UpdateStockByStockSellerCodeRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***stockItems.stockItem.sellerStockCode*** | *Ürün stok mağaza kodu* |
| ***stockItems.stockItem.quantity*** | *Yeni ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**\*Bold alanlar zorunlu alanlardır.**

***UpdateStockByStockSellerCodeResponse***

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Stok tanımının olduğu seçenek adı* |
| *stockItems.stockItem.attributes.atribute.value* | *Stok tanımının olduğu seçenek değeri* |
| *stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *stockItems.stockItem.sellerStockCode* | *Ürün stok mağaza kodu* |
| *stockItems.stockItem.stock* | *Ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**UpdateStockByStockSellerStockCode Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:UpdateStockByStockSellerCodeRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<sellerStockCode>StokKod1234567</sellerStockCode>

<quantity>20</quantity>

<version>1</version>

</stockItem>

</stockItems>

</sch:UpdateStockByStockSellerCodeRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:UpdateStockByStockSellerCodeResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<stockItems>

<stockItem>

<attributes>

<attribute>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<id>122375331751</id>

<sellerStockCode>StokKod1234567</sellerStockCode>

<stock>20</stock>

<version>2</version>

</stockItem>

</stockItems>

</ns3:UpdateStockByStockSellerCodeResponse>

</env:Body>

</env:Envelope>

**UpdateStockByStockSellerStockCode Örnek Çağrı (JAVA)**

public static void main (String [] args) {   
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 String strSellerStockCode = "StokKod1234567";  
 int quantityValue = 10;  
 long versionValue = 0;  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
 StockItemForUpdateStockWithSellerStockCodeList stockItemList = new StockItemForUpdateStockWithSellerStockCodeList ();  
 StockItemForUpdateStockWithSellerStockCode stockItem = new StockItemForUpdateStockWithSellerStockCode();  
 stockItem.setVersion(versionValue);  
 stockItem.setSellerStockCode(strSellerStockCode);  
 stockItem.setQuantity(BigInteger.valueOf(quantityValue));  
 stockItemList.getStockItem().add(stockItem);  
  
 UpdateStockByStockSellerCodeRequest request = new UpdateStockByStockSellerCodeRequest();  
 request.setAuth(authentication);  
 request.setStockItems(stockItemList);  
  
 ProductStockServicePort port = new ProductStockServicePortService().getProductStockServicePortSoap11();  
 UpdateStockByStockSellerCodeResponse response = port.updateStockByStockSellerCode(request);  
 System.out.println("Update status is " + response.getResult().getStatus().getValue());

### Stok Özellikleri ile Stok Miktarı Artırma (IncreaseStockByStockAttributes)

Bir ürünün stok seçenek bilgilerini kullanarak stok miktarını arttırmak için kullanılır.

N11 tarafında değişen stok miktarlarını ezmemek için, “**version**” bilgisi verilmesi durumunda ilgili ürün stok bilgisinin N11 de versiyonu ile karşılaştırma yapılır, stok versiyon numaraları uyumsuz ise işlem gerçekleştirilmez.

***IncreaseStockByStockAttributesRequest***

|  |  |
| --- | --- |
| ***auth*** | [*Bkz:: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***product.id*** | *Ürünün n11 ID si* |
| ***product.stockItems.stockItem.attributes.attribute.name*** | *Stok tanımının olduğu seçenek adı* |
| ***product.stockItems.stockItem.attributes.attribute.value*** | *Stok tanımının olduğu seçenek değeri* |
| ***product.stockItems.stockItem.quantityToIncrease*** | *Arttıralacak stok miktarı* |
| *product.stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**\*Bold alanlar zorunlu alanlardır.**

***IncreaseStockByStockAttributesResponse***

|  |  |
| --- | --- |
| *result* | [*Bkz:: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Stok tanımının olduğu seçenek adı* |
| *stockItems.stockItem.attributes.atribute.value* | *Stok tanımının olduğu seçenek değeri* |
| *stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *stockItems.stockItem.sellerStockCode* | *Ürün stok mağaza kodu* |
| *stockItems.stockItem.stock* | *Ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**IncreaseStockByStockAttributes Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:IncreaseStockByStockAttributesRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<product>

<id>258512546</id>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<attributes>

<!--1 or more repetitions:-->

<attribute>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<quantityToIncrease>15</quantityToIncrease>

<version>3</version>

</stockItem>

</stockItems>

</product>

</sch:IncreaseStockByStockAttributesRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:IncreaseStockByStockAttributesResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<stockItems>

<stockItem>

<attributes>

<attribute>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<id>122375331751</id>

<sellerStockCode>StokKod1234567</sellerStockCode>

<stock>25</stock>

<version>4</version>

</stockItem>

</stockItems>

</ns3:IncreaseStockByStockAttributesResponse>

</env:Body>

</env:Envelope>

**IncreaseStockByStockAttributes Örnek Çağrı (JAVA)**

public static void main (String [] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

long productIdValue = 258512546L;

String strStockAttName = "Renk";

String strStockAttValue = "Kırmızı";

int quantityIncreaseValue = 10;

long versionValue = 5;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

StockAttributeList stockAttributeList = new StockAttributeList();

StockAttribute stockAttribute = new StockAttribute();

stockAttribute.setName(strStockAttName);

stockAttribute.setValue(strStockAttValue);

stockAttributeList.getAttribute().add(stockAttribute);

StockItemForAddStockWithAttributesList stockItemList = new StockItemForAddStockWithAttributesList();

StockItemForAddStockWithAttributes stockItem = new StockItemForAddStockWithAttributes();

stockItem.setVersion(versionValue);

stockItem.setAttributes(stockAttributeList);

stockItem.setQuantityToIncrease(BigInteger.valueOf(quantityIncreaseValue));

stockItemList.getStockItem().add(stockItem);

ProductWithIdAndStockAttributes productWithIdAndStockAttributes = new ProductWithIdAndStockAttributes();

productWithIdAndStockAttributes.setId(productIdValue);

productWithIdAndStockAttributes.setStockItems(stockItemList);

IncreaseStockByStockAttributesRequest request = new IncreaseStockByStockAttributesRequest();

request.setAuth(authentication);

request.setProduct(productWithIdAndStockAttributes);

ProductStockServicePort port = new ProductStockServicePortService().getProductStockServicePortSoap11();

IncreaseStockByStockAttributesResponse response = port.increaseStockByStockAttributes(request);

System.out.println("Increasing status is " + response.getResult().getStatus().getValue());

}

### N11 Ürün Stok ID ile Stok Miktarı Artırma (IncreaseStockByStockId)

Bir ürünün n11 ürün stok ID bilgisini kullanarak stok miktarını arttırmak için kullanılır. Bir ürüne ait n11 ürün stok ID sine, **ProductStockService** içindeki **GetProductStockByProductId** veya **GetProductStockBySellerCode** metotları ile ulaşılabilir.

N11 tarafında değişen stok miktarlarını ezmemek için, “**version**” bilgisi verilmesi durumunda ilgili ürün stok bilgisinin N11 de versiyonu ile karşılaştırma yapılır, stok versiyon numaraları uyumsuz ise işlem gerçekleştirilmez.

***IncreaseStockByStockIdRequest***

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***stockItems.stockItem.id*** | *Ürün stok id bilgisi* |
| ***stockItems.stockItem.quantityToIncrease*** | *Arttıralacak stok miktarı* |
| *product.stockItems.stockItem.version* | *Ürün versiyon numarası* |

**\*Bold alanlar zorunlu alanlardır.**

***IncreaseStockByStockIdResponse***

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Stok tanımının olduğu seçenek adı* |
| *stockItems.stockItem.attributes.atribute.value* | *Stok tanımının olduğu seçenek değeri* |
| *stockItems.stockItem.id* | *Ürün stokitem N11 ID si* |
| *stockItems.stockItem.sellerCode* | *Ürün stok mağaza kodu* |
| *stockItems.stockItem.quantity* | *Ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**IncreaseStockByStockId Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:IncreaseStockByStockIdRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<id>122375331751</id>

<quantityToIncrease>5</quantityToIncrease>

<version>6</version>

</stockItem>

</stockItems>

</sch:IncreaseStockByStockIdRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:IncreaseStockByStockIdResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<stockItems>

<stockItem>

<attributes>

<attribute>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<id>122375331751</id>

<sellerStockCode>StokKod1234567</sellerStockCode>

<stock>60</stock>

<version>7</version>

</stockItem>

</stockItems>

</ns3:IncreaseStockByStockIdResponse>

</env:Body>

</env:Envelope>

**IncreaseStockByStockId Örnek Çağrı (JAVA)**

public static void main (String [] args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 long stockItemIdValue = 122375331751L;  
 int quantityIncreaseValue = 10;  
 long versionValue = 7;  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 StockItemForAddStockWithIdList stockList = new StockItemForAddStockWithIdList();  
 StockItemForAddStockWithId stockItem = new StockItemForAddStockWithId();  
 stockItem.setQuantityToIncrease(BigInteger.valueOf(quantityIncreaseValue));  
 stockItem.setId(stockItemIdValue);  
 stockItem.setVersion(versionValue);  
 stockList.getStockItem().add(stockItem);  
  
 IncreaseStockByStockIdRequest request = new IncreaseStockByStockIdRequest();  
 request.setAuth(authentication);  
 request.setStockItems(stockList);  
  
 ProductStockServicePort port = new ProductStockServicePortService().getProductStockServicePortSoap11();  
 IncreaseStockByStockIdResponse response = port.increaseStockByStockId(request);  
  
 System.out.println("Increasing status is " + response.getResult().getStatus().getValue());  
  
}

### Mağaza Stok Kodu ile Stok Miktarı Arttırma (IncreaseStockByStockSellerCode)

Bir ürünün mağaza stok kodu kullanarak stok miktarını arttırmak için kullanılır.

N11 tarafında değişen stok miktarlarını ezmemek için, “**version**” bilgisi verilmesi durumunda ilgili ürün stok bilgisinin N11 de versiyonu ile karşılaştırma yapılır, stok versiyon numaraları uyumsuz ise işlem gerçekleştirilmez.

***IncreaseStockByStockSellerCodeRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***stockItems.stockItem.sellerStockCode*** | *Mağaza stok kodu* |
| ***stockItems.stockItem.quantityToIncrease*** | *Arttıralacak stok miktarı* |
| *product.stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**\*Bold alanlar zorunlu alanlardır.**

***IncreaseStockByStockSellerCodeResponse***

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Stok tanımının olduğu seçenek adı* |
| *stockItems.stockItem.attributes.atribute.value* | *Stok tanımının olduğu seçenek değeri* |
| *stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *stockItems.stockItem.sellerStockCode* | *Ürün stok mağaza kodu* |
| *stockItems.stockItem.stock* | *Ürün stok miktarı* |
| *stockItems.stockItem.version* | *Ürün stok versiyon numarası* |

**IncreaseStockByStockSellerCode Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:IncreaseStockByStockSellerCodeRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<stockItems>

<!--1 or more repetitions:-->

<stockItem>

<sellerStockCode>StokKod1234567</sellerStockCode>

<quantityToIncrease>5</quantityToIncrease>

<version>8</version>

</stockItem>

</stockItems>

</sch:IncreaseStockByStockSellerCodeRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:IncreaseStockByStockSellerCodeResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<stockItems>

<stockItem>

<attributes>

<attribute>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<id>122375331751</id>

<sellerStockCode>StokKod1234567</sellerStockCode>

<stock>90</stock>

<version>11</version>

</stockItem>

</stockItems>

</ns3:IncreaseStockByStockSellerCodeResponse>

</env:Body>

</env:Envelope>

**IncreaseStockByStockSellerCode Örnek Çağrı (JAVA)**

public static void main (String [] args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 String strSellerStockCode = "StokKod1234567";  
 int quantityIncreaseValue = 10;  
 long versionValue = 9;  
  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 StockItemForAddStockWithSellerStockCodeList stockList = new StockItemForAddStockWithSellerStockCodeList();  
 StockItemForAddStockWithSellerStockCode stockItem = new StockItemForAddStockWithSellerStockCode();  
 stockItem.setVersion(versionValue);  
 stockItem.setQuantityToIncrease(BigInteger.valueOf(quantityIncreaseValue));  
 stockItem.setSellerStockCode(strSellerStockCode);  
 stockList.getStockItem().add(stockItem);  
  
 IncreaseStockByStockSellerCodeRequest request = new IncreaseStockByStockSellerCodeRequest();  
 request.setAuth(authentication);  
 request.setStockItems(stockList);  
  
 ProductStockServicePort port = new ProductStockServicePortService().getProductStockServicePortSoap11();  
 IncreaseStockByStockSellerCodeResponse response = port.increaseStockByStockSellerCode(request);  
  
 System.out.println("Increasing status is " + response.getResult().getStatus().getValue());  
  
}

## 3.6 Sipariş Servisleri (OrderService)

Sipariş servisleri için WSDL adresi: <https://api.n11.com/ws/OrderService.wsdl>

### Detaylı Sipariş Listeleme (DetailedOrderList)

Verilen arama kriterlerine göre sipariş bilgisi ile beraber sipariş maddelerini de listelemek için kullanılır.

**DetailedOrderListRequest**

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| *searchData.productId* | *Ürün n11 ID si* |
| *searchData.status* | [*Bkz: SearchData*](#_Sipariş_Durumu_(SearchData)) |
| *searchData.buyerName* | *Siparişi veren N11 üye adı* |
| *searchData.orderNumber* | *Sipariş numarası* |
| *searchData.productSellerCode* | *Ürünün mağaza kodu* |
| *searchData.recipient* | *Teslim alacak alıcının adı* |
| *searchData.period.startDate* | *Sipariş oluşturma tarihi başlangıç* |
| *searchData.period.endDate* | *Sipariş oluşturma tarihi bitiş* |
| *pagingData* | [*Bkz: Sayfalandırma*](#_Sayfalandırma_(Pagination)_1) |

**\*Bold alanlar zorunlu alanlardır.**

**DetailedOrderListResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *pagingData* | [*Bkz: Sayfalandırma*](#_Sayfalandırma_(Pagination)_1) |
| *orderList.order.citizenshipId* | *Alıcı TC kimlik numarası* |
| *orderList.order.createDate* | *Sipariş oluşturma tarihi* |
| *orderList.order.id* | *Sipariş N11 ID si* |
| *orderList.order.orderNumber* | *Sipariş numarası* |
| *paymentType* | [*Bkz: Payment Type*](#_Ödeme_Tipi_(Payment) |
| *orderList.order.totalAmount* | *Siparişin toplam tutarı* |
| *orderList.order.totalDiscountAmount* | *Siparişin toplam indirim tutarı* |
| *orderList.order.orderItemList.orderItem* | *Sipariş maddeleri listesi* |
| *orderList.order.orderItemList.orderItem.commission* | *Sipariş maddesinin n11 hizmet tutarı* |
| *deliveryFeeType* | [*Bkz: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderList.order.orderItemList.orderItem.dueAmount* | *Tahsil edilecek tutar* |
| *orderList.order.orderItemList.orderItem.id* | *Sipariş maddesi ID si* |
| *orderList.order.orderItemList.orderItem.installmentChargeWithVAT* | *Vade farkı(kdv dahil)* |
| *orderList.order.orderItemList.orderItem.mallDiscount* | *Ürün ile ilgili n11 indirimi* |
| *orderList.order.orderItemList.orderItem.price* | *Ürünün birim fiyatı (indirimler hariç)* |
| *orderList.order.orderItemList.orderItem.productId* | *Ürün n11 ID si* |
| *orderList.order.orderItemList.orderItem.productName* | *Ürün adı* |
| *orderList.order.orderItemList.orderItem.productSellerCode* | *Ürün mağaza kodu* |
| *orderList.order.orderItemList.orderItem.quantity* | *Sipariş adedi* |
| *orderList.order.orderItemList.orderItem.sellerCouponDiscount* | *Mağaza kupon indirimi* |
| *orderList.order.orderItemList.orderItem.sellerDiscount* | *Mağaza indirimi* |
| *orderList.order.orderItemList.orderItem.sellerInvoiceAmount* | *Mağaza fatura tutarı* |
| *orderList.order.orderItemList.orderItem.shipmentInfo.shipmentCompany.id* | *Kargo şirket id bilgisi* |
| *orderList.order.orderItemList.orderItem.shipmentInfo.shipmentCompany.name* | *Kargo şirket adı* |
| *orderList.order.orderItemList.orderItem.shipmentInfo.shipmentMethod* | *Kargo yöntemi 1 ise kargo 2 ise diğer* |
| *orderList.order.orderItemList.orderItem.shipmentDate* | *Kargo tarihi* |
| *orderItem.status* | [*Bkz: OrderItem Status*](#_Sipariş_Maddesi_Durum) |
| *orderList.order.orderItemList.orderItem.totalMallDiscountPrice* | *N11 indirimleri toplamı* |
| *orderList.order.orderItemList.orderItem.version* | *Sipariş kaleminin stok takip versiyon numarası* |
| *orderList.order.orderNumber* | *Sipariş numarası* |
| *orderList.order.paymentType* | [*Bkz: Ödeme tipi*](#_Ödeme_Tipi__1) |
| *orderList.order.status* | [*Bkz: Sipariş Durumu*](#_Sipariş_Durum_Bilgisi) |
| *orderList.order.totalAmount* | *Sipariş tutarı* |
| *orderList.order.totalDiscountAmount* | *Sipariş toplam indirim tutarı (Satıcı+n11)* |

**DetailedOrderList Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:DetailedOrderListRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<searchData>

<productId>258512546</productId>

<status>1</status>

<buyerName></buyerName>

<orderNumber></orderNumber>

<productSellerCode></productSellerCode>

<recipient></recipient>

<sameDayDelivery></sameDayDelivery>

<period>

<startDate>15/09/2016</startDate>

<endDate>15/09/2016</endDate>

</period>

</searchData>

<pagingData>

<currentPage>0</currentPage>

<pageSize>99</pageSize>

<totalCount>99</totalCount>

<pageCount>99</pageCount>

</pagingData>

</sch:DetailedOrderListRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:DetailedOrderListResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result><status>success</status></result>

<pagingData>

<currentPage>0</currentPage>

<pageSize>99</pageSize>

<totalCount>1</totalCount>

<pageCount>1</pageCount>

</pagingData>

<orderList>

<order>

<citizenshipId>\*\*\*\*\*\*\*\*\*\*\*</citizenshipId>

<createDate>15/09/2016 13:29</createDate>

<id>113975650</id>

<orderItemList>

<orderItem>

<approvedDate/>

<attributes>

<attribute>

<id>354081125</id>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<commission>8.25</commission>

<customTextOptionValues/>

<deliveryFeeType>2</deliveryFeeType>

<dueAmount>55.00</dueAmount>

<id>18675500</id>

<installmentChargeWithVAT>0</installmentChargeWithVAT>

<mallDiscount>0</mallDiscount>

<price>55</price>

<productId>258512546</productId>

<productName>My Test Product Title</productName>

<productSellerCode>MySellerCode1234</productSellerCode>

<quantity>1</quantity>

<sellerCouponDiscount>0</sellerCouponDiscount>

<sellerDiscount>0</sellerDiscount>

<sellerInvoiceAmount>55</sellerInvoiceAmount>

<sellerStockCode>StokKod1234567</sellerStockCode>

<shipmentInfo>

<campaignNumberStatus>2</campaignNumberStatus>

<shipmentCode>18011950</shipmentCode>

<shipmentCompany>

<id>389</id>

<name>Balnak</name>

<shortName>BLK</shortName>

</shipmentCompany>

<shipmentMethod>1</shipmentMethod>

</shipmentInfo>

<shippingDate/>

<status>2</status>

<totalMallDiscountPrice>0.00</totalMallDiscountPrice>

<version>12</version>

</orderItem>

</orderItemList>

<orderNumber>qa202125281764</orderNumber>

<paymentType>1</paymentType>

<status>2</status>

<totalAmount>55</totalAmount>

<totalDiscountAmount>0</totalDiscountAmount>

</order>

</orderList>

</ns3:DetailedOrderListResponse>

</env:Body>

</env:Envelope>

**DetailedOrderList Örnek Çağrı (JAVA)**

public static void main (String [] args ) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strStartDate = "15.09.2016";

String strEndDate = "15.09.2016";

String strOrderStatus = "1";

String strRecipient = "";

String strBuyerName = "";

String strOrderNumber = "";

String strProductSellerCode = "";

long productIdValue = 258512546L;

long totalCountValue = 10;

int currentPageValue = 0;

int pageCountValue = 1;

int pageSizeValue = 10;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

OrderSearchPeriod orderSearchPeriod = new OrderSearchPeriod();

orderSearchPeriod.setStartDate(strStartDate);

orderSearchPeriod.setEndDate(strEndDate);

OrderDataListRequest orderDataListRequest = new OrderDataListRequest();

orderDataListRequest.setProductSellerCode(strProductSellerCode);

orderDataListRequest.setRecipient(strRecipient);

orderDataListRequest.setPeriod(orderSearchPeriod);

orderDataListRequest.setBuyerName(strBuyerName);

orderDataListRequest.setProductId(productIdValue);

orderDataListRequest.setOrderNumber(strOrderNumber);

orderDataListRequest.setStatus(strOrderStatus);

PagingData pagingData = new PagingData();

pagingData.setCurrentPage(currentPageValue);

pagingData.setPageCount(pageCountValue);

pagingData.setPageSize(pageSizeValue);

pagingData.setTotalCount(totalCountValue);

DetailedOrderListRequest request = new DetailedOrderListRequest();

request.setAuth(authentication);

request.setPagingData(pagingData);

request.setSearchData(orderDataListRequest);

OrderServicePort port = new OrderServicePortService().getOrderServicePortSoap11();

DetailedOrderListResponse response = port.detailedOrderList(request);

List<DetailedOrderData> orderList = response.getOrderList().getOrder();

for (DetailedOrderData sampleOrder:orderList

) {

System.out.println("Order ID: " + sampleOrder.getId());

}

}

### Sipariş Listeleme (OrderList)

*Bu metot sipariş ile ilgili özet bilgileri listelemek için kullanılır.*

***OrderListRequest***

|  |  |
| --- | --- |
| ***auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| *searchData.productId* | *Ürün n11 ID si* |
| *searchData.status* | [*Bkz: SearchData*](#_Sipariş_Durumu_(SearchData)) |
| *searchData.buyerName* | *Siparişi veren N11 üye adı* |
| *searchData.orderNumber* | *Sipariş numarası* |
| *searchData.productSellerCode* | *Ürünün mağaza kodu* |
| *searchData.recipient* | *Teslim alacak alıcının adı* |
| *searchData.period.startDate* | *Sipariş oluşturma tarihi başlangıç* |
| *searchData.period.endDate* | *Sipariş oluşturma tarihi bitiş* |
| *pagingData* | [*Bkz: Sayfalandırma*](#_Sayfalandırma_(Pagination)_1) |

**\*Bold alanlar zorunlu alanlardır.**

**OrderListResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderList.order.citizenshipId* | *TC kimlik numarası* |
| *orderList.order.createDate* | *Sipariş oluşturma tarihi* |
| *orderList.order.id* | *Sipariş n11 ID bilgisi* |
| *orderList.order.orderNumber* | *Sipariş numarası* |
| *paymentType* | [*Bkz: Payment Type*](#_Ödeme_Tipi_(Payment) |
| *orderList.order.status* | [*Bkz: Sipariş Durumu*](#_Sipariş_Durum_Bilgisi_1) |

**OrderList Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:OrderListRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<searchData>

<productId></productId>

<status>1</status>

<buyerName></buyerName>

<orderNumber></orderNumber>

<productSellerCode></productSellerCode>

<recipient></recipient>

<sameDayDelivery></sameDayDelivery>

<period>

<startDate></startDate>

<endDate></endDate>

</period>

</searchData>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

</pagingData>

</sch:OrderListRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:OrderListResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

<totalCount>4</totalCount>

<pageCount>1</pageCount>

</pagingData>

<orderList>

<order>

<citizenshipId>\*\*\*\*\*\*\*\*\*\*\*</citizenshipId>

<createDate>20/07/2016 17:32</createDate>

<id>113914001</id>

<orderNumber>qa206576291768</orderNumber>

<paymentType>1</paymentType>

<status>5</status>

</order>

<order>

<citizenshipId>>\*\*\*\*\*\*\*\*\*\*\*</</citizenshipId>

<createDate>28/07/2016 10:36</createDate>

<id>113922397</id>

<orderNumber>qa202927691760</orderNumber>

<paymentType>1</paymentType>

<status>5</status>

</order>

<order>

<citizenshipId>>\*\*\*\*\*\*\*\*\*\*\*</</citizenshipId>

<createDate>15/09/2016 13:29</createDate>

<id>113975650</id>

<orderNumber>qa202125281764</orderNumber>

<paymentType>1</paymentType>

<status>2</status>

</order>

<order>

<citizenshipId>>\*\*\*\*\*\*\*\*\*\*\*</</citizenshipId>

<createDate>15/09/2016 13:49</createDate>

<id>113975651</id>

<orderNumber>qa206125281761</orderNumber>

<paymentType>1</paymentType>

<status>2</status>

</order>

</orderList>

</ns3:OrderListResponse>

</env:Body>

</env:Envelope>

**OrderList Örnek Çağrı (JAVA)**

public static void main (String [] args ) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strStartDate = "15.09.2016";

String strEndDate = "15.09.2016";

String strOrderStatus = "1";

String strRecipient = "";

String strBuyerName = "";

String strOrderNumber = "";

String strProductSellerCode = "";

long productIdValue = 0;

int currentPageValue = 0;

int pageSizeValue = 10;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

OrderSearchPeriod orderSearchPeriod = new OrderSearchPeriod();

orderSearchPeriod.setStartDate(strStartDate);

orderSearchPeriod.setEndDate(strEndDate);

OrderDataListRequest orderDataListRequest = new OrderDataListRequest();

orderDataListRequest.setProductSellerCode(strProductSellerCode);

orderDataListRequest.setRecipient(strRecipient);

orderDataListRequest.setPeriod(orderSearchPeriod);

orderDataListRequest.setBuyerName(strBuyerName);

orderDataListRequest.setProductId(productIdValue);

orderDataListRequest.setOrderNumber(strOrderNumber);

orderDataListRequest.setStatus(strOrderStatus);

RequestPagingData pagingData = new RequestPagingData();

pagingData.setCurrentPage(currentPageValue);

pagingData.setPageSize(pageSizeValue);

OrderListRequest request = new OrderListRequest();

request.setAuth(authentication);

request.setPagingData(pagingData);

request.setSearchData(orderDataListRequest);

OrderServicePort port = new OrderServicePortService().getOrderServicePortSoap11();

OrderListResponse response = port.orderList(request);

List<OrderData> orderList = response.getOrderList().getOrder();

for (OrderData sampleOrder:orderList

) {

System.out.println("Order ID: " + sampleOrder.getId());

}

}

### Sipariş Detayı (OrderDetail)

*Sipariş N11 ID bilgisi kullanarak sipariş detaylarını almak için kullanılır, sipariş N11 ID bilgisine OrderService OrderList veya DetailedOrderList metotlarıyla ulaşılabilir.*

*n11 platform üzerinden kargo ücretinin ödenmesi ve bunun tahsilat bilgileri “serviceItemList” alanından ulaşılabilir.*

***OrderDetailRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***orderRequest.id*** | *Sipariş N11 ID si* |

**\*Bold alanlar zorunlu alanlardır.**

**OrderDetailResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderDetail.citizenshipId* | *TC kimlik numarası* |
| *orderDetail.createDate* | *Sipariş oluşturma tarihi* |
| *orderDetail.id* | *Sipariş N11 ID si* |
| *orderDetail.orderNumber* | *Sipariş numarası* |
| *paymentType* | [*Bkz: Payment Type*](#_Ödeme_Tipi_) |
| *orderDetail.billingAddress.address* | *Fatura adres bilgisi* |
| *orderDetail.billingAddress.city* | *Adres şehri* |
| *orderDetail.billingAddress.district* | *Adres ilçesi* |
| *orderDetail.billingAddress.fullName* | *Adres adı* |
| *orderDetail.billingAddress.gsm* | *Alıcı telefon numarası* |
| *orderDetail.billingAddress.neighborhood* | *Adres semt/mahallesi* |
| *orderDetail.billingAddress.postalCode* | *Adres posta kodu* |
| *orderDetail.billingTemplate.dueAmount* | *Vade farkı* |
| *ordelDetail.billingTemplate.installmentChargeWithVat* | *Vade farkı kdv dahil* |
| *orderDetail.billingTemplate.originalPrice* | *Sipariş Kaleminin bütün indirimlerden önceki fiyatı* |
| *orderDetail.billingTemplate.sellerInvoiceAmount* | *Komisyon fatura tutarı* |
| *orderDetail.billingTemplate.totalMallDiscountPrice* | *Tüm N11 indirimleri* |
| *orderDetail.billingTemplate.totalSellerDiscount* | *Toplam Satıcı indirimi* |
| *orderDetail.billingTemplate.totalServiceItemOriginalPrice* | *Toplam Servis Fiyatı* |
| *orderDetail.buyer.fullName* | *Siparişi veren alıcı adı* |
| *orderDetail.buyer.id* | *Siparişi veren alıcı N11 ID si* |
| *orderDetail.buyer.taxId* | *Vergi id numarası* |
| *orderDetail.buyer.taxOffice* | *Vergi dairesi* |
| *orderDetail.buyer.email* | *Siparişi veren alıcı emaili* |
| *orderDetail.invoiceType* | *Fatura tipi: 1: Bireysel*  *2: Kurumsal* |
| *orderDetail.itemList.item.approveDate* | *Sipariş kalemi kabul edildiği tarih* |
| *orderDetail.itemList.item.attributes.attribute.name* | *Sipariş kalemi özellik adı* |
| *orderDetail.itemList.item.attributes.attribute.value* | *Sipariş kalemi özellik değeri* |
| *orderDetail.itemList.item.commission* | *Sipariş maddesinin n11 hizmet tutarı* |
| *orderDetail.itemList.item.customTextOptionsValues* | *Müşterinin sipariş kalemi için girdiği ek bilgi* |
| *orderDetail.itemList.item.sellerInvoiceAmount* | *Satıcı fatura tutarı(satıcının bir sipariş için kesmesi gereken fatura tutarı)* |
| *orderDetail.itemList.item.deliveryFeeType* | [*Bkz: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderDetail.itemList.item.dueAmount* | *Tahsil edilecek tutar* |
| *orderDetail.itemList.item.id* | *Sipariş maddesi n11 ID si* |
| *orderDetail.itemList.item.installmentChargeWithVAT* | *Vade farkı (KDV dahil)* |
| *orderDetail.itemList.item.mallDiscount* | *Ürün ile ilgili n11 indirimi* |
| *orderDetail.itemList.item.price* | *Ürünün birim fiyatı (liste fiyatı, indirimler hariç)* |
| *orderDetail.itemList.item.productId* | *Ürün n11 ID si* |
| *orderDetail.itemList.item.productName* | *Ürün adı* |
| *orderDetail.itemList.item.productSellerCode* | *Ürün mağaza kodu* |
| *orderDetail.itemList.item.quantity* | *Sipariş adedi* |
| *orderDetail.itemList.item.sellerCouponDiscount* | *Mağaza kupon indirimi* |
| *orderDetail.itemList.item.sellerDiscount* | *Mağaza indirimi* |
| *orderDetail.itemList.item.shipmentInfo.campaignNumberStatus* | *Kampanya numarası durumu;*  *1:USED(Kullanılmış) 2: UnUSED (Kullanılmamış)* |
| *orderDetail.itemList.item.shipmentInfo.shipmentCode* | *Kargo kodu* |
| *orderDetail.itemList.item.shipmentInfo.shipmentCompany.id* | *Kargo şirket id bilgisi* |
| *orderDetail.itemList.item.shipmentInfo.shipmentCompany.name* | *Kargo şirket adı* |
| *orderDetail.itemList.item.shipmentInfo.shipmentCompany.shortName* | *Kargo şirketi kısa adı* |
| *orderDetail.itemList.item.shipmentInfo.shipmentMethod* | *Kargo yöntemi 1: Kargo 2: Diğer* |
| *orderDetail.itemList.item.shipmentDate* | *Kargolama tarihi* |
| *orderDetail.itemList.item.status* | [*Bkz: OrderItem Status*](#_Sipariş_Maddesi_Durum) |
| *orderDetail.itemList.item.totalMallDiscountPrice* | *n11 indirimleri toplamı* |
| *orderDetail.itemList.item.version* | *İlgili ürünün stok versiyon numarası* |
| *orderDetail.serviceItemList.serviceItem.intallmentChargeWithVAT* | *Servis kdv dahil vade farkı* |
| *orderDetail.serviceItemList.serviceItem.orderItemList.orderItem* | *Servis sipariş kalem idsi* |
| *orderDetail.serviceItemList.serviceItem.orderItemType* | [*Bkz: Servis sipariş kalem tipi*](#_Sipariş_Kalemi_tipi) |
| *orderDetail.serviceItemList.serviceItem.price* | *Servis kalemi fiyatı* |
| *orderDetail.serviceItemList.serviceItem.quantity* | *Servis adedi* |
| *orderDetail.serviceItemList.serviceItem.sellerInvoiceAmount* | *Servis fatura tutarı* |
| *orderDetail.serviceItemList.serviceItem.totalDiscountPrice* | *Servis toplam indirim tutarı* |
| *orderDetail.shippingAddress.address* | *Kargo adres bilgisi* |
| *orderDetail.shippingAddress.city* | *Kargo adres şehri* |
| *orderDetail.shippingAddress.district* | *Kargo adres semti* |
| *orderDetail.shippingAddress.fullName* | *Kargo adres adı* |
| *orderDetail.shippingAddress.gsm* | *Kargo adres tel* |
| *orderDetail.shippingAddress.neighborhood* | *Kargo semt/mahallesi* |
| *orderDetail.shippingAddress.postalCode* | *Kargo adres posta kodu* |

**OrderDetail Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:OrderDetailRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<orderRequest>

<id>113975650</id>

</orderRequest>

</sch:OrderDetailRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:OrderDetailResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<orderDetail>

<citizenshipId>\*\*\*\*\*\*\*\*\*\*\*</citizenshipId>

<createDate>15/09/2016 13:29</createDate>

<id>113975650</id>

<orderNumber>202125281764</orderNumber>

<paymentType>1</paymentType>

<status>2</status>

<billingAddress>

<address>Örnek Mahallesi Örnek Sokak No:0</address>

<city>İstanbul</city>

<district>Sarıyer</district>

<fullName>\*\*\*\*\* \*\*\*\*\*</fullName>

<gsm>53\*\*\*\*\*\*\*\*</gsm>

</billingAddress>

<billingTemplate>

<dueAmount>55.00</dueAmount>

<installmentChargeWithVat>0</installmentChargeWithVat>

<originalPrice>55</originalPrice>

<sellerInvoiceAmount>55</sellerInvoiceAmount>

<totalMallDiscountPrice>0.00</totalMallDiscountPrice>

<totalSellerDiscount>0</totalSellerDiscount>

<totalServiceItemOriginalPrice>0</totalServiceItemOriginalPrice>

</billingTemplate>

<buyer>

<email>b\*\*\*\*\*\*\*\*\*\*\*.a\*\*\*\*@n11.com</email>

<fullName>test user</fullName>

<id>4625057</id>

<taxId/>

<taxOffice/>

<tcId>\*\*\*\*\*\*\*\*\*\*\*</tcId>

</buyer>

<invoiceType>1</invoiceType>

<itemList>

<item>

<approvedDate/>

<attributes>

<attribute>

<id>354081125</id>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<commission>8.25</commission>

<customTextOptionValues/>

<deliveryFeeType>2</deliveryFeeType>

<dueAmount>55.00</dueAmount>

<id>18675500</id>

<installmentChargeWithVAT>0</installmentChargeWithVAT>

<mallDiscount>0</mallDiscount>

<price>55</price>

<productId>258512546</productId>

<productName>My Test Product Title</productName>

<productSellerCode>MySellerCode1234</productSellerCode>

<quantity>1</quantity>

<sellerCouponDiscount>0</sellerCouponDiscount>

<sellerDiscount>0</sellerDiscount>

<sellerInvoiceAmount>55</sellerInvoiceAmount>

<sellerStockCode>StokKod1234567</sellerStockCode>

<shipmentInfo>

<campaignNumberStatus>2</campaignNumberStatus>

<shipmentCode>18011950</shipmentCode>

<shipmentCompany>

<id>389</id>

<name>Balnak</name>

<shortName>BLK</shortName>

</shipmentCompany>

<shipmentMethod>1</shipmentMethod>

</shipmentInfo>

<shippingDate/>

<status>2</status>

<totalMallDiscountPrice>0.00</totalMallDiscountPrice>

<version>12</version>

</item>

</itemList>

<serviceItemList/>

<shippingAddress>

<address>Deneme</address>

<city>İstanbul</city>

<district>Sarıyer</district>

<fullName>B\*\*\*\*\*\* A\*\*\* A\*\*\*\*</fullName>

<gsm>53\*\*\*\*\*\*\*\*</gsm>

</shippingAddress>

</orderDetail>

</ns3:OrderDetailResponse>

</env:Body>

</env:Envelope>

**OrderDetail Örnek Çağrı (JAVA)**

public static void main(String[] argv) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

long orderIdValue = 113975650;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

OrderDataRequest orderDataRequest = new OrderDataRequest();

orderDataRequest.setId(orderIdValue);

OrderDetailRequest request = new OrderDetailRequest();

request.setAuth(authentication);

request.setOrderRequest(orderDataRequest);

OrderServicePort port = new OrderServicePortService().getOrderServicePortSoap11();

OrderDetailResponse orderDetailResponse = port.orderDetail(request);

OrderDetailData orderDetail = orderDetailResponse.getOrderDetail();

System.out.println("Order number: " + orderDetail.getOrderNumber());

List<OrderSearchData> orderItemList = orderDetail.getItemList().getItem();

for (OrderSearchData sampleItem:orderItemList

) {

System.out.println("Order item ID(s): " + sampleItem.getId());

}

}

### Sipariş Maddesinin Kabul Edilmesi (OrderItemAccept)

Sipariş maddesinin n11 ID si kullanılarak kabul edilmesi için kullanılır. Kabul edilen sipariş daha sonra mağaza tarafından reddedilemez. Sipariş n11 ID sine **OrderService** içinden **OrderDetail** veya **DetailedOrderList** metodu kullanılarak erişilir.

***OrderItemAcceptRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***orderItem.id*** | *Sipariş kalemi n11 ID si* |

**\*Bold alanlar zorunlu alanlardır.**

***OrderItemAcceptResponse***

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderItemList.orderItem.aprovedDate* | *Siparişin onaylanma tarihi* |
| *orderItemList.orderItem.attributes.attribute.name* | *Sipariş kalemi özellik adı* |
| *orderItemList.orderItem.attributes.attribute.value* | *Sipariş kalemi özellik değeri* |
| *orderItemList.orderItem.commission* | *Sipariş maddesi ile ilgili n11 hizmet tutarı* |
| *orderItemList.orderItem.customTextOptionValues* | *Sipariş kalemine girilen özellik değeri* |
| *orderItemList.orderItem.deliveryFeeType* | [*Bkz: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderItemList.orderItem.dueAmount* | *Tahsil edilecek tutar* |
| *orderItemList.orderItem.id* | *Sipariş maddesi n11 ID bilgisi* |
| *orderItemList.orderItem.installmentChargeWithVAT* | *Vade farkı(kdv dahil)* |
| *orderItemList.orderItem.mallDiscount* | *Ürün ile ilgili n11 indirimi* |
| *orderItemList.orderItem.price* | *Ürünün birim fiyatı (indirimler hariç)* |
| *orderItemList.orderItem.productId* | *Ürünün n11 ID si* |
| *orderItemList.orderItem.productName* | *Ürün adı* |
| *orderItemList.orderItem.productSellerCode* | *Ürün mağaza kodu* |
| *orderItemList.orderItem.quantity* | *Sipariş adedi* |
| *orderItemList.orderItem.sellerCouponDiscount* | *Mağaza kupon indirimi* |
| *orderItemList.orderItem.sellerDiscount* | *Mağaza indirimi* |
| *orderItemList.orderItem.sellerInvoiceAmount* | *Mağaza fatura tutarı* |
| *orderItemList.orderitem.shipmentInfo.campaignNumber* | *Kampanya Numarası* |
| *orderItemList.orderitem.shipmentInfo.campaignNumberStatus* | *Kampanya Numara Durumu*  *1: Used 2:Unused* |
| *orderItemList.orderitem.shipmentInfo.shipmentCode* | *Kargo kodu* |
| *orderItemList.orderitem.shipmentInfo.shipmentCompany.id* | *Kargo şirket id bilgisi* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.name* | *Kargo şirket adı* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.shortName* | *Kargo şirketi kısa adı* |
| *orderItem.shipmentInfo.shipmentMethod* | *Kargo yöntemi*  *1: Kargo*  *2: Diğer* |
| *orderItem.shipmentDate* | *Kargolama tarihi* |
| *orderItem.status* | [*Bkz: OrderItem Status*](#_Sipariş_Maddesi_Durum) |
| *orderItem.totalMallDiscountPrice* | *N11 indirimleri toplamı* |
| *orderItem.version* | *İlgili ürünün stok versiyon numarası* |

**OrderItemAccept Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:OrderItemAcceptRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<orderItemList>

<orderItem>

<id>18675500</id>

</orderItem>

</orderItemList>

</sch:OrderItemAcceptRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:OrderItemAcceptResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<orderItemList>

<orderItem>

<approvedDate>15/09/2016</approvedDate>

<attributes>

<attribute>

<id>354081125</id>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<commission>8.25</commission>

<customTextOptionValues/>

<deliveryFeeType>2</deliveryFeeType>

<dueAmount>55.00</dueAmount>

<id>18675500</id>

<installmentChargeWithVAT>0</installmentChargeWithVAT>

<mallDiscount>0</mallDiscount>

<price>55</price>

<productId>258512546</productId>

<productName>My Test Product Title</productName>

<productSellerCode>MySellerCode1234</productSellerCode>

<quantity>1</quantity>

<sellerCouponDiscount>0</sellerCouponDiscount>

<sellerDiscount>0</sellerDiscount>

<sellerInvoiceAmount>55</sellerInvoiceAmount>

<sellerStockCode>StokKod1234567</sellerStockCode>

<shipmentInfo>

<campaignNumberStatus>2</campaignNumberStatus>

<shipmentCode>18011950</shipmentCode>

<shipmentCompany>

<id>389</id>

<name>Balnak</name>

<shortName>BLK</shortName>

</shipmentCompany>

<shipmentMethod>1</shipmentMethod>

</shipmentInfo>

<shippingDate/>

<status>5</status>

<totalMallDiscountPrice>0.00</totalMallDiscountPrice>

<version>12</version>

</orderItem>

</orderItemList>

</ns3:OrderItemAcceptResponse>

</env:Body>

</env:Envelope>

**OrderItemAccept Örnek Çağrı (JAVA)**

public static void main(String[] argv) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

long orderItemIdVal = 18675501;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

OrderItemDataRequest orderItemDataRequest = new OrderItemDataRequest();

orderItemDataRequest.setId(orderItemIdVal);

OrderItemDataListRequest orderItemDataListRequest = new OrderItemDataListRequest();

orderItemDataListRequest.getOrderItem().add(orderItemDataRequest);

OrderItemAcceptRequest request = new OrderItemAcceptRequest();

request.setAuth(authentication);

request.setOrderItemList(orderItemDataListRequest);

OrderServicePort port = new OrderServicePortService().getOrderServicePortSoap11();

OrderItemAcceptResponse response = port.orderItemAccept(request);

List<OrderItemData> itemDataList = response.getOrderItemList().getOrderItem();

}

### Sipariş Kaleminin Reddedilmesi (OrderItemReject)

Sipariş maddesinin n11 ID si kullanılarak reddedilmesi için kullanılır. Reddedilen sipariş daha sonra mağaza tarafından kabul edilemez. Sipariş n11 ID sine **OrderService** içinden **OrderDetail** veya **DetailedOrderList** metodu kullanılarak erişilir.

***OrderItemRejectRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***orderItemList.orderItem.id*** | *Sipariş maddesi n11 ID bilgisi* |
| ***rejectReason*** | *Sipariş reddetme sebebi* |
| ***rejectReasonType*** | *Sipariş reddetme tipi :*  *OUT\_OF\_STOCK (Stokta yok), OTHER(Diğer)* |

**\*Bold alanlar zorunlu alanlardır.**

**OrderItemRejectResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderItemList.orderItem.approvedDate* | *Siparişin onaylanma tarihi* |
| *orderItemList.orderItem.attributes* | *Sipariş kaleminin özellikleri* |
| *orderItemList.orderItem.commission* | *Sipariş maddesi ile ilgili n11 hizmet tutarı* |
| *orderItemList.orderItem.customTextOptionsValues* | *Müşterinin sipariş kalemi için girdiği ek bilgi* |
| *orderItemList.orderItem.deliveryFeeType* | [*Bkz: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderItemList.orderItem.dueAmount* | *Tahsil edilecek tutar* |
| *orderItemList.orderItem.id* | *Sipariş maddesi n11 ID bilgisi* |
| *orderItemList.orderItem.installmentChargeWithVAT* | *Vade farkı (kdv dahil)* |
| *orderItemList.orderItem.mallDiscount* | *Ürün ile ilgili n11 indirimi* |
| *orderItemList.orderItem.price* | *Ürünün birim fiyatı (indirimler hariç)* |
| *orderItemList.orderItem.productId* | *Ürünün n11 ID si* |
| *orderItemList.orderItem.productName* | *Ürün adı* |
| *orderItemList.orderItem.productSellerCode* | *Ürün mağaza kodu* |
| *orderItemList.orderItem.quantity* | *Sipariş adedi* |
| *orderItemList.orderitem.rejectReason* | *Sipariş reddetme sebebi* |
| *orderItemList.orderitem.rejectReasonType* | *Sipariş reddetme tipi* |
| *orderItemList.orderItem.sellerCouponDiscount* | *Mağaza kupon indirimi* |
| *orderItemList.orderItem.sellerDiscount* | *Mağaza indirimi* |
| *orderItemList.orderItem.sellerInvoiceAmount* | *Mağaza fatura tutarı* |
| *orderItemList.orderItem.shipmentInfo.campaignNumberStatus* | *Kampanya numarası durumu;*  *1:USED(Kullanılmış) 2: UnUSED (Kullanılmamış)* |
| *orderItemList.orderItemshipmentInfo.shipmentCode* | *Kargo kodu* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.id* | *Kargo şirket id bilgisi* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.name* | *Kargo şirket adı* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.shortName* | *Kargo şirketi kısa adı* |
| *orderItemList.orderItem.shipmentInfo.shipmentMethod* | *Kargo yöntemi*  *1: Kargo*  *2: Diğer* |
| *orderItemList.orderItem.shippingDate* | *Kargolama tarihi* |
| *orderItemList.orderItem.status* | [*Bkz: Orderıtem Status*](#_Sipariş_Maddesi_Durum) |
| *orderItemList.orderItem.totalMallDiscountPrice* | *N11 indirimleri toplamı* |
| *orderItemList.orderItem.version* | *İlgili ürünün stok versiyon numarası* |

**OrderItemReject Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:OrderItemRejectRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<orderItemList>

<!--1 or more repetitions:-->

<orderItem>

<id>18675502</id>

</orderItem>

</orderItemList>

<rejectReason>Ürün stoklarımızda bulunmamaktadır.</rejectReason>

<rejectReasonType>OUT\_OF\_STOCK</rejectReasonType>

</sch:OrderItemRejectRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:OrderItemRejectResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<orderItemList>

<orderItem>

<approvedDate/>

<attributes>

<attribute>

<name>Seçenek 1</name>

<value>Değer 1</value>

</attribute>

<attribute>

<name>Seçenek 2</name>

<value>Değer 2</value>

</attribute>

</attributes>

<commission>0.00</commission>

<customTextOptionValues/>

<deliveryFeeType>2</deliveryFeeType>

<dueAmount>0.00</dueAmount>

<id>18675502</id>

<installmentChargeWithVAT>0.00</installmentChargeWithVAT>

<mallDiscount>0</mallDiscount>

<price>30</price>

<productId>258491034</productId>

<productName>Ürünün başlığı</productName>

<productSellerCode>URUN-1</productSellerCode>

<quantity>0</quantity>

<rejectReason>Ürün stoklarımızda bulunmamaktadır.</rejectReason>

<rejectReasonType>OUT\_OF\_STOCK</rejectReasonType>

<sellerCouponDiscount>0.00</sellerCouponDiscount>

<sellerDiscount>0</sellerDiscount>

<sellerInvoiceAmount>0.00</sellerInvoiceAmount>

<shipmentInfo>

<campaignNumberStatus>2</campaignNumberStatus>

<shipmentCode>18011952</shipmentCode>

<shipmentCompany>

<id>389</id>

<name>Balnak</name>

<shortName>BLK</shortName>

</shipmentCompany>

<shipmentMethod>1</shipmentMethod>

</shipmentInfo>

<shippingDate/>

<status>8</status>

<totalMallDiscountPrice>0.00</totalMallDiscountPrice>

<version>2</version>

</orderItem>

</orderItemList>

</ns3:OrderItemRejectResponse>

</env:Body>

</env:Envelope>

**OrderItemReject Örnek Çağrı (JAVA)**

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strRejectReason = "Ürün stoklarımızda bulunmamaktadır.";

String strRejReasonType = "OUT\_OF\_STOCK";

long orderItemIdVal = 18675503;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

OrderItemDataRequest orderItemDataRequest = new OrderItemDataRequest();

orderItemDataRequest.setId(orderItemIdVal);

OrderItemDataRequestForRejection orderItemDataRequestForRejection = new OrderItemDataRequestForRejection();

orderItemDataRequestForRejection.getOrderItem().add(orderItemDataRequest);

OrderItemRejectRequest request = new OrderItemRejectRequest();

request.setAuth(authentication);

request.setOrderItemList(orderItemDataRequestForRejection);

request.setRejectReason(strRejectReason);

request.setRejectReasonType(strRejReasonType);

OrderServicePort port = new OrderServicePortService().getOrderServicePortSoap11();

OrderItemRejectResponse response = port.orderItemReject(request);

List<OrderItemData> orderItemList = response.getOrderItemList().getOrderItem();

}

### Sipariş Kaleminin Kargoya Verilmesi (MakeOrderItemShipment)

Sipariş maddesinin n11 ID si kullanılarak kargoya verilmesi için kullanılır. Sipariş n11 ID sine **OrderService** içinden **OrderDetail** veya **DetailedOrderList** metodu kullanılarak erişilir.

Ön koşul olarak güncelleme yapılmak istenen sipariş maddesinin durumunun (OrderItemStatus) “**Ödendi**” veya “**Kabul edildi**” olması gerekmektedir. Aksi durumda “**ön koşullar sağlanamadı**” cevabı alınır.

Kargo şirketlerinin listesi için **ShipmentCompanyService** den **GetShipmentCompanies** metodu kullanılmalıdır.

**MakeOrderItemShipmentRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz: Yetkilendirme*](#_Yetkilendirme_(Authentication)) |
| ***orderItemList.orderItem.id*** | *Ürün n11 ID si* |
| ***orderItemList.orderItem.shipmentInfo.shipmentCompany.id*** | *Kargo şirketinin n11 ID si* |
| ***orderItemList.orderItem.shipmentInfo.campaignNumber*** | *Sipariş kampanya numarası* |
| ***orderItemList.orderItem.shipmentInfo.trackingNumber*** | *Takip numarası* |
| ***orderItemList.orderItem.shipmentInfo.shipmentMethod*** | *Kargo yöntemi*  *1: Kargo*  *2: Diğer* |

**\*Bold alanlar zorunlu alanlardır.**

**MakeOrderItemShipmentResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderItemList.orderItem.approvedDate* | *Siparişin onaylanma tarihi* |
| *orderItemList.orderItem.attributes* | *Sipariş kalemi özellileri* |
| *orderItemList.orderItem.commission* | *Sipariş maddesi ile ilgili n11 hizmet tutarı* |
| *orderItemList.orderItem.customTextOptionValues* | *Müşterinin sipariş kalemi için girdiği ek bilgi* |
| *orderItemList.orderItem.deliveryFeeType* | [*Bkz: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderItemList.orderItem.dueAmount* | *Tahsil edilecek tutar* |
| *orderItemList.orderItem.id* | *Sipariş maddesi n11 ID bilgisi* |
| *orderItemList.orderItem.installmentChargeWithVAT* | *Vade farkı(kdv dahil)* |
| *orderItemList.orderItem.mallDiscount* | *Ürün ile ilgili n11 indirimi* |
| *orderItemList.orderItem.price* | *Ürünün birim fiyatı (indirimler hariç)* |
| *orderItemList.orderItem.productId* | *Ürünün n11 ID si* |
| *orderItemList.orderItem.productName* | *Ürün adı* |
| *orderItemList.orderItem.productSellerCode* | *Ürün mağaza kodu* |
| *orderItemList.orderItem.quantity* | *Sipariş adedi* |
| *orderItemList.orderItem.sellerCouponDiscount* | *Mağaza kupon indirimi* |
| *orderItemList.orderItem.sellerDiscount* | *Mağaza indirimi* |
| *orderItemList.orderItem.sellerInvoiceAmount* | *Mağaza fatura tutarı* |
| *orderItemList.orderItem.shipmentInfo.campaignNumberStatus* | *Kampanya numarası durumu;*  *1:USED(Kullanılmış) 2: UnUSED (Kullanılmamış)* |
| *orderItemList.orderItem.shipmentInfo.shipmentCode* | *Kargo kodu* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.id* | *Kargo şirket id bilgisi* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.name* | *Kargo şirket adı* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.shortName* | *Kargo Şirketi kısa adı* |
| *orderItemList.orderItem.shipmentInfo.shipmentMethod* | *Kargo yöntemi*  *1: Kargo*  *2: Diğer* |
| *orderItemList.orderItem.shipmentInfo.trackingNumber* | *Takip numarası* |
| *orderItemList.orderItem.shippingDate* | *Kargolama tarihi* |
| *orderItemList.orderItem.status* | [*Bkz: OrderItem Status*](#_Sipariş_Maddesi_Durum) |
| *orderItemList.orderItem.totalMallDiscountPrice* | *N11 indirimleri toplamı* |
| *orderItemList.orderItem.version* | *İlgili ürünün stok versiyon numarası* |

**MakeOrderItemShipment Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:MakeOrderItemShipmentRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<orderItemList>

<!--1 or more repetitions:-->

<orderItem>

<id>18675500</id>

<shipmentInfo>

<shipmentCompany>

<id>341</id>

</shipmentCompany>

<campaignNumber>466805582805015</campaignNumber>

<trackingNumber>123</trackingNumber>

<shipmentMethod>1</shipmentMethod>

</shipmentInfo>

</orderItem>

</orderItemList>

</sch:MakeOrderItemShipmentRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:MakeOrderItemShipmentResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<orderItemList>

<orderItem>

<approvedDate>15/09/2016</approvedDate>

<attributes>

<attribute>

<id>354081125</id>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<commission>8.25</commission>

<customTextOptionValues/>

<deliveryFeeType>2</deliveryFeeType>

<dueAmount>55.00</dueAmount>

<id>18675500</id>

<installmentChargeWithVAT>0</installmentChargeWithVAT>

<mallDiscount>0</mallDiscount>

<price>55</price>

<productId>258512546</productId>

<productName>My Test Product Title</productName>

<productSellerCode>MySellerCode1234</productSellerCode>

<quantity>1</quantity>

<sellerCouponDiscount>0</sellerCouponDiscount>

<sellerDiscount>0</sellerDiscount>

<sellerInvoiceAmount>55</sellerInvoiceAmount>

<sellerStockCode>StokKod1234567</sellerStockCode>

<shipmentInfo>

<campaignNumberStatus>2</campaignNumberStatus>

<shipmentCode>18011950</shipmentCode>

<shipmentCompany>

<id>341</id>

<name>Sürat</name>

<shortName>SRT</shortName>

</shipmentCompany>

<shipmentMethod>1</shipmentMethod>

<trackingNumber>123</trackingNumber>

</shipmentInfo>

<shippingDate>15/09/2016</shippingDate>

<status>6</status>

<totalMallDiscountPrice>0.00</totalMallDiscountPrice>

<version>12</version>

</orderItem>

</orderItemList>

</ns3:MakeOrderItemShipmentResponse>

</env:Body>

</env:Envelope>

**MakeOrderItemShipment Örnek Çağrı (JAVA)**

public static void main(String[] argv) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strCampaignNumber = "466805582805015";

String strTrackingNumber = "123";

long orderItemIdVal = 18675505;

int shipmentCompanyIdVal = 341;

int shipmentMethodValue = 1;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

ShipmentCompanyRequest shipmentCompanyRequest = new ShipmentCompanyRequest();

shipmentCompanyRequest.setId(shipmentCompanyIdVal);

MakeShipmentInfoRequest makeShipmentInfoRequest = new MakeShipmentInfoRequest();

makeShipmentInfoRequest.setCampaignNumber(strCampaignNumber);

makeShipmentInfoRequest.setTrackingNumber(strTrackingNumber);

makeShipmentInfoRequest.setShipmentMethod(BigInteger.valueOf(shipmentMethodValue));

makeShipmentInfoRequest.setShipmentCompany(shipmentCompanyRequest);

OrderItemShipmentRequest orderItemShipmentRequest = new OrderItemShipmentRequest();

orderItemShipmentRequest.setId(orderItemIdVal);

orderItemShipmentRequest.setShipmentInfo(makeShipmentInfoRequest);

OrderItemListShipmentRequest orderItemListShipmentRequest = new OrderItemListShipmentRequest();

orderItemListShipmentRequest.getOrderItem().add(orderItemShipmentRequest);

MakeOrderItemShipmentRequest makeOrderItemShipmentRequest = new MakeOrderItemShipmentRequest();

makeOrderItemShipmentRequest.setAuth(authentication);

makeOrderItemShipmentRequest.setOrderItemList(orderItemListShipmentRequest);

OrderServicePort port = new OrderServicePortService().getOrderServicePortSoap11();

MakeOrderItemShipmentResponse response = port.makeOrderItemShipment(makeOrderItemShipmentRequest);

System.out.println(response.getResult().getStatus().getValue());

}

## 3.7 Kargo Şirketi Servisi (ShipmentCompanyService)

Kargo şirketi servisi için WSDL adresi:

<https://api.n11.com/ws/ShipmentCompanyService.wsdl>

### Kargo Şirketlerini Listeleme (GetShipmentCompanies)

Sistemde tanımlı olan tüm kargo şirketlerini listeler.

Genel kullanıma açık bir servis olduğu için servisin kullanımı sırasında herhangi bir güvenlik kontrolü yapılmamaktadır.

**GetShipmentCompaniesResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *shipmentCompanies.shipmentCompany.id* | *Kargo şirket id bilgisi* |
| *shipmentCompanies.shipmentCompany.name* | *Kargo şirket adı* |
| *shipmentCompanies.shipmentCompany.shortName* | *Kargo şirket kısa adı* |

**GetShipmentCompanies Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetShipmentCompaniesRequest/>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetShipmentCompaniesResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<shipmentCompanies>

<shipmentCompany>

<id>27050</id>

<name>Aynı Gün Teslimat</name>

<shortName>AKF</shortName>

</shipmentCompany>

<shipmentCompany>

<id>341</id>

<name>Sürat</name>

<shortName>SRT</shortName>

</shipmentCompany>

<shipmentCompany>

<id>344</id>

<name>Yurtiçi</name>

<shortName>YK</shortName>

</shipmentCompany>

...

</shipmentCompanies>

</ns3:GetShipmentCompaniesResponse>

</env:Body>

</env:Envelope>

**GetShipmentCompanies Örnek Çağrı (JAVA)**

public static void main(String[] args) {

GetShipmentCompaniesRequest request = new GetShipmentCompaniesRequest();

ShipmentCompanyServicePort port = new ShipmentCompanyServicePortService().getShipmentCompanyServicePortSoap11();

GetShipmentCompaniesResponse response = port.getShipmentCompanies(request);

List<ShipmentCompanyData> shipmentCompanyList = response.getShipmentCompanies().getShipmentCompany();

for (ShipmentCompanyData sampleCompany:shipmentCompanyList

) {

System.out.println(sampleCompany.getId() + " " + sampleCompany.getName() + " " + sampleCompany.getShortName());

}

}

## [3.8 Teslimat Şablonu Servisi (ShipmentService)](#_3.7_Kargolama_Servisi)

Teslimat şablonu servisi için WSDL adresi: <https://api.n11.com/ws/ShipmentService.wsdl>

### [Teslimat Şablonu Listeleme (GetShipmentTemplate)](#_Kargo_Şirketlerini_Listeleme)

Teslimat şablon ismi ile aratılan şablonun adres metod gibi özelliklerini gösterme. deliverableCities teslimat yapılacak şehirlerin seçimini yaptığımız alan bu alana değer girilmezse tüm şehirlere gönderim yapılacak anlamındadır.

**GetShipmentTemplateRequest**

|  |  |
| --- | --- |
| ***Auth*** | *Bkz:* [*authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***name*** | *Kargo şablonu adı* |

**GetShipmentTemplateResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *shipmentTemplate.claimShipmentCompany.name* | *İade/Değişim kargo şirket adı* |
| *shipmentTemplate.claimShipmentCompany.shortName* | *İade/Değişim kargo şirket kısa adı* |
| *shipmentTemplate.combinedShipmentAllowed* | [*Bkz: Shipment Allowed*](#_Kargo_Çoklu_Gönderim) |
| *shipmentTemplate.deliverableCities.city.code* | *Teslimat yapılacak şehir kodu* |
| *shipmentTemplate.deliverableCities.city.name* | *Teslimat yapılacak şehir adı* |
| *shipmentTemplate.deliveryFeeType* | *Kargo ödeme yöntemi 1 ise alıcı öder 2 ise mağaza öder 3 ise şartlı kargo 4 ise N11 öder* |
| *shipmentTemplate.exchangeAddress.address* | *İade adres açıklaması* |
| *shipmentTemplate.exchangeAddress.city.code* | *Iade adres şehir kodu* |
| *shipmentTemplate.exchangeAddress.city.name* | *Iade adres şehir ismi* |
| *shipmentTemplate.exchangeAddress.district.id* | *Iade adres bölge idsi* |
| *shipmentTemplate.exchangeAddress.district.name* | *Iade adres bölge adı* |
| *shipmentTemplate.exchangeAddress.postalCode* | *Iade adres posta kodu* |
| *shipmentTemplate.exchangeAddress.title* | *Iade adres başlık adı* |
| *shipmentTemplate.exchangeAddress.feeCondition* | *Koşullu kargonun koşul tipi :*  *1:ByPrice 2: ByUnit* |
| *shipmentTemplate.exchangeAddress.feeConditionPrice/feeConditionUnit* | *Koşullu kargonun koşul fiyatı/miktarı* |
| *shipmentTemplate.exchangeInfo* | *Değişim içeriği* |
| *shipmentTemplate.shipmentCompanies.shipmentCompany.name* | *Kargo şirket adı* |
| *shipmentTemplate.shipmentCompanies.shipmentCompany.shortName* | *Kargo şirket kısa adı* |
| *shipmentTemplate.shipmentMethod* | *Ürünün ulaşma şekli 1ise kargo 2 ise diğer* |
| *shipmentTemplate.shippingInfo* | *Teslimat bilgileri/text bilgisi girilecek* |
| *shipmentTemplate.specialDelivery* | *Özel teslimat true/false* |
| *shipmentTemplate.templateName* | *Kargo şablonu* |
| *shipmentTemplate.warehouseAddess.address* | *Depo adres bilgisi* |
| *shipmentTemplate.warehouseAddess.district.name* | *Depo ilçe adı* |
| *shipmentTemplate.warehouseAddess.district.id* | *Depo adres ilçe idsi* |
| *shipmentTemplate.warehouseAddess.feeCondition* | *Koşullu kargonun koşul tipi :*  *1:ByPrice 2: ByUnit* |
| *shipmentTemplate.warehouseAddess.feeConditionPrice* | *Koşullu kargonun koşul fiyatı/miktarı* |
| *shipmentTemplate.warehouseAddess.city.name* | *Depo adres şehir adı* |
| *shipmentTemplate.warehouseAddess.city.code* | *Depo adres şehir kodu* |
| *shipmentTemplate.warehouseAddess.city.postalCode* | *Depo adres posta kodu* |
| *shipmentTemplate.warehouseAddess.title* | *Depo adres başlığı* |

**GetShipmentTemplate Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetShipmentTemplateRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<name>Şablon</name>

</sch:GetShipmentTemplateRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetShipmentTemplateResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<shipmentTemplate>

<claimShipmentCompany>

<name>Sürat</name>

<shortName>SRT</shortName>

</claimShipmentCompany>

<combinedShipmentAllowed>true</combinedShipmentAllowed>

<deliverableCities>

<city>

<code>1</code>

<name>Adana</name>

</city>

<city>

<code>2</code>

<name>Adıyaman</name>

</city>

<city>

<code>3</code>

<name>Afyon</name>

</city>

...

<city>

<code>77</code>

<name>Yalova</name>

</city>

<city>

<code>66</code>

<name>Yozgat</name>

</city>

<city>

<code>67</code>

<name>Zonguldak</name>

</city>

</deliverableCities>

<deliveryFeeType>1</deliveryFeeType>

<exchangeAddress>

<address>Deneme Sokak No:0</address>

<city>

<code>1</code>

<name>Adana</name>

</city>

<district>

<id>22695</id>

<name>Ceyhan</name>

</district>

<postalCode>01001</postalCode>

<title>Ev</title>

</exchangeAddress>

<shipmentCompanies>

<shipmentCompany>

<name>Balnak</name>

<shortName>BLK</shortName>

</shipmentCompany>

<shipmentCompany>

<name>Horoz Lojistik</name>

<shortName>HL</shortName>

</shipmentCompany>

</shipmentCompanies>

<shipmentMethod>1</shipmentMethod>

<specialDelivery>false</specialDelivery>

<templateName>Şablon</templateName>

<useDmallCargo>false</useDmallCargo>

<warehouseAddress>

<address>Deneme Bulvarı No:0</address>

<city>

<code>1</code>

<name>Adana</name>

</city>

<district>

<id>22695</id>

<name>Ceyhan</name>

</district>

<postalCode>01001</postalCode>

<title>Ev</title>

</warehouseAddress>

</shipmentTemplate>

</ns3:GetShipmentTemplateResponse>

</env:Body>

</env:Envelope>

**GetShipmentTemplate Örnek Çağrı (JAVA)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strShipmentName = "Şablon";

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

GetShipmentTemplateRequest getShipmentTemplateRequest = new GetShipmentTemplateRequest();

getShipmentTemplateRequest.setAuth(authentication);

getShipmentTemplateRequest.setName(strShipmentName);

ShipmentServicePort port = new ShipmentServicePortService().getShipmentServicePortSoap11();

GetShipmentTemplateResponse response = port.getShipmentTemplate(getShipmentTemplateRequest);

List<ShipmentCompanyApiModel> shipmentCompanyList = response.getShipmentTemplate().getShipmentCompanies().getShipmentCompany();

for (ShipmentCompanyApiModel sampleShipmentCompany:shipmentCompanyList

) {

System.out.println(sampleShipmentCompany.getName() + " " + sampleShipmentCompany.getShortName());}}

### [Teslimat Şablonu Oluşturma, Güncelleme (CreateOrUpdateShipmentTemplate)](#_Kargo_Şirketlerini_Listeleme)

Teslimat şablonu kargonun nasıl gideceğine dair oluşturulan bir şablondur.Siparişlerde kullanılacak olan teslimat şablonu özellikleriyle birlikte bu servis aracılığı ile oluşturulur. claimshipmentCompany <useDmallCargo> alanı kullan yani “1” ise zorunludur.

**CreateOrUpdateShipmentTemplateRequest**

|  |  |
| --- | --- |
| ***Auth*** | *Bkz:* [*authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***shipment.templateName*** | *Kargo şablonu* |
| ***shipment.installmentInfo*** | *Kurulum bilgileri/text bilgisi girilecek* |
| ***shipment.exchangeInfo*** | *Değişim bilgileri/text bilgisi girilecek* |
| ***shipment.shippingInfo*** | *Teslimat bilgileri/text bilgisi girilecek* |
| ***shipment.specialDelivery*** | *Özel teslimat true/false* |
| ***shipment.deliveryFeeType*** | *Kargo ödeme yöntemi 1 ise alıcı öder 2 ise mağaza öder 3 ise şartlı kargo* |
| *shipment.combinedShipmentAllowed* | [*Bkz: Shipment Allowed*](#_Kargo_Çoklu_Gönderim) |
| ***shipment.shipmentMethod*** | *Ürünün ulaşma şekli 1ise kargo 2 ise diğer* |
| ***shipment.warehouseAddess.title*** | *Depo adres başlığı* |
| ***shipment.warehouseAddess.address*** | *Depo adres bilgisi* |
| ***shipment.warehouseAddess.district.name*** | *Depo ilçe adı* |
| ***shipment.warehouseAddess.district.id*** | *Depo adres ilçe idsi* |
| ***shipment.warehouseAddess.city.name*** | *Depo adres şehir adı* |
| ***shipment.warehouseAddess.city.code*** | *Depo adres şehir kodu* |
| ***shipment.warehouseAddess.city.postalCode*** | *Depo adres posta kodu* |
| ***shipment.exchangeAddress.address*** | *İade adres açıklaması* |
| ***shipment.exchangeAddress.city.code*** | *Iade adres şehir kodu* |
| ***shipment.exchangeAddress.city.name*** | *Iade adres şehir ismi* |
| ***shipment.exchangeAddress.district.id*** | *Iade adres bölge idsi* |
| ***shipment.exchangeAddress.district.name*** | *Iade adres bölge adı* |
| ***shipment.exchangeAddress.postalCode*** | *Iade adres posta kodu* |
| ***shipment.exchangeAddress.title*** | *Iade adres başlık adı* |
| ***shipment.shipmentCompanies.shipmentCompany.name*** | *Kargo şirket adı* |
| ***shipment.shipmentCompanies.shipmentCompany.shortName*** | *Kargo şirket kısa adı* |
| *shipmentTemplate.deliverableCities.city.code* | *Teslimat yapılacak şehir kodu* |
| *shipmentTemplate.deliverableCities.city.name* | *Teslimat yapılacak şehir adı* |
| ***shipmentTemplate.claimShipmentCompany.name*** | *İade/Değişim kargo şirket adı* |
| ***shipmentTemplate.claimShipmentCompany.shortName*** | *İade/Değişim kargo şirket kısa adı* |
| *shipment.cargoAccountNo* | *Kargo hesap no* |
| ***shipment.useDmallCargo*** | *n11.com entegrasyonlu kargo firmasi kullanımı:*  *1: kullan*  *0: kullanma* |

**\*Bold alanlar zorunlu alanlardır.**

**CreateOrUpdateShipmentTemplateResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *shipmentTemplate.claimShipmentCompany.name* | *İade/Değişim kargo şirket adı* |
| *shipmentTemplate.claimShipmentCompany.shortName* | *İade/Değişim kargo şirket kısa adı* |
| *shipmentTemplate.combinedShipmentAllowed* | [*Bkz: Shipment Allowed*](#_Kargo_Çoklu_Gönderim) |
| *shipmentTemplate.deliverableCities.city.code* | *Teslimat yapılacak şehir kodu* |
| *shipmentTemplate.deliverableCities.city.name* | *Teslimat yapılacak şehir adı* |
| *shipmentTemplate.deliveryFeeType* | [*Bkz: Deliver Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *shipmentTemplate.exchangeAddress.address* | *İade adres açıklaması* |
| *shipmentTemplate.exchangeAddress.city.code* | *Iade adres şehir kodu* |
| *shipmentTemplate.exchangeAddress.city.name* | *Iade adres şehir ismi* |
| *shipmentTemplate.exchangeAddress.district.id* | *Iade adres bölge idsi* |
| *shipmentTemplate.exchangeAddress.district.name* | *Iade adres bölge adı* |
| *shipmentTemplate.exchangeAddress.postalCode* | *Iade adres posta kodu* |
| *shipmentTemplate.exchangeAddress.title* | *Iade adres başlık adı* |
| *shipmentTemplate.exchangeInfo* | *Değişim bilgileri/text bilgisi girilecek* |
| *shipmentTemplate.installmentInfo* | *Kurulum bilgileri/text bilgisi girilecek* |
| *shipmentTemplate.shipmentCompanies.shipmentCompany.name* | *Kargo şirket adı* |
| *shipmentTemplate.shipmentCompanies.shipmentCompany.shortName* | *Kargo şirket kısa adı* |
| *shipmentTemplate.shipmentMethod* | *Ürünün ulaşma şekli 1ise kargo 2 ise diğer* |
| *shipmentTemplate.shippingInfo* | *Teslimat bilgileri/text bilgisi girilecek* |
| *shipmentTemplate.specialDelivery* | *Özel teslimat true/false* |
| *shipmentTemplate.templateName* | *Kargo şablonu* |
| *shipmentTemplate.useDmallCargo* | *n11.com entegrasyonlu kargo firmasi kullanımı ;*  *1: kullan*  *0: kullanma* |
| *shipmentTemplate.warehouseAddess.title* | *Depo adres başlığı* |
| *shipmentTemplate.warehouseAddess.address* | *Depo adres bilgisi* |
| *shipmentTemplate.warehouseAddess.district.name* | *Depo ilçe adı* |
| *shipmentTemplate.warehouseAddess.district.id* | *Depo adres ilçe idsi* |
| *shipmentTemplate.warehouseAddess.city.name* | *Depo adres şehir adı* |
| *shipmentTemplate.warehouseAddess.city.code* | *Depo adres şehir kodu* |
| *shipmentTemplate.warehouseAddess.city.postalCode* | *Depo adres posta kodu* |

**CreateOrUpdateShipmentTemplate Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:CreateOrUpdateShipmentTemplateRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

<shipment>

<templateName>Yeni Adres</templateName>

<installmentInfo>Kurulum</installmentInfo>

<exchangeInfo>Değişim bilgileri</exchangeInfo>

<shippingInfo>İyi teslimat iyi kargo</shippingInfo>

<specialDelivery>true</specialDelivery>

<deliveryFeeType>1</deliveryFeeType>

<combinedShipmentAllowed>true</combinedShipmentAllowed>

<shipmentMethod>1</shipmentMethod>

<warehouseAddress>

<title>İş</title>

<address>Deneme Caddesi Deneme Sokak No:0</address>

<city>

<code>4</code>

<name>Ağrı</name>

</city>

<district>

<id>22249</id>

<name>Hamur</name>

</district>

<postalCode>35474</postalCode>

</warehouseAddress>

<exchangeAddress>

<title>Değişim</title>

<address>Değiştim Bulvarı No:0</address>

<city>

<code>4</code>

<name>Ağrı</name>

</city>

<district>

<id>22249</id>

<name>Hamur</name>

</district>

<postalCode>35474</postalCode>

</exchangeAddress>

<shipmentCompanies>

<!--1 or more repetitions:-->

<shipmentCompany>

<name>Yurtiçi</name>

<shortName>YK</shortName>

</shipmentCompany>

</shipmentCompanies>

<claimShipmentCompany>

<name>Yurtiçi</name>

<shortName>YK</shortName>

</claimShipmentCompany>

<useDmallCargo>1</useDmallCargo>

</shipment>

</sch:CreateOrUpdateShipmentTemplateRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:CreateOrUpdateShipmentTemplateResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<shipmentTemplate>

<claimShipmentCompany>

<name>Yurtiçi</name>

<shortName>YK</shortName>

</claimShipmentCompany>

<combinedShipmentAllowed>true</combinedShipmentAllowed>

<deliverableCities/>

<deliveryFeeType>1</deliveryFeeType>

<exchangeAddress>

<address>Değiştim Bulvarı No:0</address>

<city>

<code>4</code>

<name>Ağrı</name>

</city>

<district>

<id>22249</id>

<name>Hamur</name>

</district>

<postalCode>35474</postalCode>

<title>Değişim</title>

</exchangeAddress>

<exchangeInfo>Değişim bilgileri</exchangeInfo>

<installmentInfo>Kurulum</installmentInfo>

<shipmentCompanies>

<shipmentCompany>

<name>Yurtiçi</name>

<shortName>YK</shortName>

</shipmentCompany>

</shipmentCompanies>

<shipmentMethod>1</shipmentMethod>

<shippingInfo>İyi teslimat iyi kargo</shippingInfo>

<specialDelivery>true</specialDelivery>

<templateName>Yeni Adres</templateName>

<useDmallCargo>false</useDmallCargo>

<warehouseAddress>

<address>Deneme Caddesi Deneme Sokak No:0</address>

<city>

<code>4</code>

<name>Ağrı</name>

</city>

<district>

<id>22249</id>

<name>Hamur</name>

</district>

<postalCode>35474</postalCode>

<title>İş</title>

</warehouseAddress>

</shipmentTemplate>

</ns3:CreateOrUpdateShipmentTemplateResponse>

</env:Body>

</env:Envelope>

**CreateOrUpdateShipmentTemplate Örnek Çağrı (JAVA)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

String strShipmentName = "Şablon";

String strShpShortName = "YK";

String strDistrictName = "Hamur";

String strCityName = "İstanbul";

String strAddress = "Deneme Bulvarı No:0/0";

String strPostalCode = "34887";

String strAddressTitle = "DenemeAdresi";

String strDeliveryFeeType = "1";

String strExchangeInfo = "1";

String strInstallmentInfo = "1";

String strShipmentMethod = "1";

String strShippingInfo = "14 gün içerisinde ücretsiz kargo ile iade edebilirsiniz.";

String strShpTemplateName = "DenemeTemplate";

int districtIdVal = 22249;

int cityCodeValue = 2;

boolean isCombinedShipmentAllowed = true;

boolean isSpecialDelivery = true;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

ShipmentCompanyApiModel shipmentCompanyApiModel = new ShipmentCompanyApiModel();

shipmentCompanyApiModel.setName(strShipmentName);

shipmentCompanyApiModel.setShortName(strShpShortName);

CityApiModelList cityApiModelList = new CityApiModelList();

cityApiModelList.getCity();

DistrictApiModel districtApiModel = new DistrictApiModel();

districtApiModel.setName(strDistrictName);

districtApiModel.setId(districtIdVal);

CityApiModel cityApiModel = new CityApiModel();

cityApiModel.setName(strCityName);

cityApiModel.setCode(BigInteger.valueOf(cityCodeValue));

ShipmentSaveAddress shipmentSaveAddress = new ShipmentSaveAddress();

shipmentSaveAddress.setAddress(strAddress);

shipmentSaveAddress.setCity(cityApiModel);

shipmentSaveAddress.setDistrict(districtApiModel);

shipmentSaveAddress.setPostalCode(strPostalCode);

shipmentSaveAddress.setTitle(strAddressTitle);

ShipmentCompanyApiModelList shipmentCompanyList = new ShipmentCompanyApiModelList();

shipmentCompanyList.getShipmentCompany().add(shipmentCompanyApiModel);

ShipmentApiModel shipmentApiModel = new ShipmentApiModel();

shipmentApiModel.setClaimShipmentCompany(shipmentCompanyApiModel);

shipmentApiModel.setCombinedShipmentAllowed(isCombinedShipmentAllowed);

shipmentApiModel.setDeliverableCities(cityApiModelList);

shipmentApiModel.setExchangeAddress(shipmentSaveAddress);

shipmentApiModel.setDeliveryFeeType(strDeliveryFeeType);

shipmentApiModel.setExchangeInfo(strExchangeInfo);

shipmentApiModel.setInstallmentInfo(strInstallmentInfo);

shipmentApiModel.setShipmentMethod(strShipmentMethod);

shipmentApiModel.setSpecialDelivery(isSpecialDelivery);

shipmentApiModel.setWarehouseAddress(shipmentSaveAddress);

shipmentApiModel.setExchangeAddress(shipmentSaveAddress);

shipmentApiModel.setShippingInfo(strShippingInfo);

shipmentApiModel.setTemplateName(strShpTemplateName);

shipmentApiModel.setShipmentCompanies(shipmentCompanyList);

CreateOrUpdateShipmentTemplateRequest createOrUpdateShipmentTemplateRequest = new CreateOrUpdateShipmentTemplateRequest();

createOrUpdateShipmentTemplateRequest.setAuth(authentication);

createOrUpdateShipmentTemplateRequest.setShipment(shipmentApiModel);

ShipmentServicePort port = new ShipmentServicePortService().getShipmentServicePortSoap11();

CreateOrUpdateShipmentTemplateResponse response = port.createOrUpdateShipmentTemplate(createOrUpdateShipmentTemplateRequest);

System.out.println("Procedure status is " + response.getResult().getStatus().getValue());

}

### Teslimat Şablonu Listeleme (GetShipmentTemplateList)

Oluşturulan teslimat şablonu bilgilerini listelemek için kullanılan metoddur.

**GetShipmentTemplateListRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*Bkz:: Authentication*](#_2.1_Güvenlik_Mekanizması) |

**\*Bold alanlar zorunlu alanlardır.**

**GetShipmentTemplateListResponse**

|  |  |
| --- | --- |
| *result* | [*Bkz: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *shipmentTemplates.shipmentTemplate.claimShipmentCompany.name* | *İade/Değişim kargo şirket adı* |
| *shipmentTemplates.shipmentTemplate.claimShipmentCompany.shortName* | *İade/Değişim kargo şirket kısa adı* |
| *shipmentTemplates.shipmentTemplate.combinedShipmentAllowed* | [*Bkz: Shipment Allowed*](#_Kargo_Çoklu_Gönderim) |
| *shipmentTemplates.shipmentTemplate.deliverableCities.city.code* | *Teslimat yapılacak şehir kodu* |
| *shipmentTemplates.shipmentTemplate.deliverableCities.city.name* | *Teslimat yapılacak şehir adı* |
| *shipmentTemplates.shipmentTemplate.deliveryFeeType* | [*Bkz: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.address* | *İade adres açıklaması* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.city.code* | *Iade adres şehir kodu* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.city.name* | *Iade adres şehir ismi* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.district.id* | *Iade adres bölge idsi* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.district.name* | *Iade adres bölge adı* |
| *shipmentTemplates .shipmentTemplate.exchangeAddress.feeCondition* | *Koşullu kargonun koşul tipi :*  *1:ByPrice 2: ByUnit* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.feeConditionPrice/feeConditionUnit* | *Koşullu kargonun koşul fiyatı/miktarı* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.postalCode* | *Iade adres posta kodu* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.title* | *Iade adres başlık adı* |
| *shipmentTemplates.shipmentTemplate.exchangeInfo* | *Değişim bilgileri/text bilgisi girilecek* |
| *shipmentTemplates.shipmentTemplate.shipmentCompanies.shipmentCompany.name* | *Kargo şirket adı* |
| *shipmentTemplates.shipmentTemplate.shipmentCompanies.shipmentCompany.shortName* | *Kargo şirket kısa adı* |
| *shipmentTemplates.shipmentTemplate.shipmentMethod* | *Ürünün ulaşma şekli 1 ise kargo 2 ise diğer* |
| *shipmentTemplates.shipmentTemplate.shippingInfo* | *Teslimat bilgileri/text bilgisi girilecek* |
| *shipmentTemplates.shipmentTemplate.specialDelivery* | *Özel teslimat true/false* |
| *shipmentTemplates.shipmentTemplate.templateName* | *Kargo şablonu* |
| *shipmentTemplates.shipmentTemplate.useDmallCargo* | *n11.com entegrasyonlu kargo firmasi kullanımı ;*  *1: kullan*  *0: kullanma* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.title* | *Depo adres başlığı* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.address* | *Depo adres bilgisi* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.district.name* | *Depo ilçe adı* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.district.id* | *Depo adres ilçe idsi* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.city.name* | *Depo adres şehir adı* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.city.code* | *Depo adres şehir kodu* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.feeCondition* | *Koşullu kargonun koşul tipi :*  *1:ByPrice 2: ByUnit* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.feeConditionPrice/feeConditionUnit* | *Koşullu kargonun koşul fiyatı/miktarı* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.city.postalCode* | *Depo adres posta kodu* |

**GetShipmentTemplateList Örnek Çağrı**

**Request**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.n11.com/ws/schemas">

<soapenv:Header/>

<soapenv:Body>

<sch:GetShipmentTemplateListRequest>

<auth>

<appKey>\*\*\*</appKey>

<appSecret>\*\*\*</appSecret>

</auth>

</sch:GetShipmentTemplateListRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Header/>

<env:Body>

<ns3:GetShipmentTemplateListResponse xmlns:ns3="http://www.n11.com/ws/schemas">

<result>

<status>success</status>

</result>

<shipmentTemplates>

<shipmentTemplate>

<claimShipmentCompany>

<name>Sürat</name>

<shortName>SRT</shortName>

</claimShipmentCompany>

<combinedShipmentAllowed>true</combinedShipmentAllowed>

<deliverableCities>

<city>

<code>1</code>

<name>Adana</name>

</city>

<city>

<code>2</code>

<name>Adıyaman</name>

</city>

<city>

<code>3</code>

<name>Afyon</name>

</city>

...

<code>77</code>

<name>Yalova</name>

</city>

<city>

<code>66</code>

<name>Yozgat</name>

</city>

<city>

<code>67</code>

<name>Zonguldak</name>

</city>

</deliverableCities>

<deliveryFeeType>1</deliveryFeeType>

<exchangeAddress>

<address>Örnek Adres</address>

<city>

<code>1</code>

<name>Adana</name>

</city>

<district>

<id>22695</id>

<name>Ceyhan</name>

</district>

<postalCode>01001</postalCode>

<title>Ev</title>

</exchangeAddress>

<shipmentCompanies>

<shipmentCompany>

<name>Balnak</name>

<shortName>BLK</shortName>

</shipmentCompany>

<shipmentCompany>

<name>Horoz Lojistik</name>

<shortName>HL</shortName>

</shipmentCompany>

</shipmentCompanies>

<shipmentMethod>1</shipmentMethod>

<specialDelivery>false</specialDelivery>

<templateName>Şablon</templateName>

<useDmallCargo>false</useDmallCargo>

<warehouseAddress>

<address>Örnek Adres</address>

<city>

<code>1</code>

<name>Adana</name>

</city>

<district>

<id>22695</id>

<name>Ceyhan</name>

</district>

<postalCode>01001</postalCode>

<title>Ev</title>

</warehouseAddress>

</shipmentTemplate>

</shipmentTemplates>

</ns3:GetShipmentTemplateListResponse>

</env:Body>

</env:Envelope>

**GetShipmentTemplateList Örnek Çağrı (JAVA)**

public static void main(String[] args) {

String strAppKey = "\*\*\*";

String strAppSecret = "\*\*\*";

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

GetShipmentTemplateListRequest request = new GetShipmentTemplateListRequest();

request.setAuth(authentication);

ShipmentServicePort port = new ShipmentServicePortService().getShipmentServicePortSoap11();

GetShipmentTemplateListResponse response = port.getShipmentTemplateList(request);

List<ShipmentApiModel> shpTemplateList = response.getShipmentTemplates().getShipmentTemplate();

for (ShipmentApiModel sampleTemplate:shpTemplateList

) {

System.out.println(sampleTemplate.getTemplateName() + " Address:" + sampleTemplate.getExchangeAddress().getAddress());

}

}

## 3.9 Genel Olarak Kullanılan Data Şablonları

### Yetkilendirme (Authentication)

|  |  |
| --- | --- |
| *auth.appKey* | *Servislere erişim için kullanılacak uygulama anahtarı* |
| *auth.appSecret* | *Servislere erişim için kullanılacak uygulama şifresi* |

### Sonuç Durum Bilgisi (Result)

|  |  |
| --- | --- |
| *result.status* | *İşlem sonucu durum bilgisini gösterir. “success” ve “failure” durum bilgilerini döner.* |

### Sayfalandırma (Pagination)

|  |  |
| --- | --- |
| *pagingData.currentPage* | *Gösterilen sayfa numarası (currentpage 0 ile başlar)* |
| *pagingData.pageSize* | *Sayfadaki madde sayısı* |
| *pagingData.totalCount\** | *Toplam madde sayısı* |
| *pagingData.pageCount\** | *Toplam sayfa sayısı* |

\*Request yaparken kullanılmaz, çağrı yapan tarafa response içinde döner.

### Ürün Özet Bilgisi (Product Summary Data)

|  |  |
| --- | --- |
| *product.currencyAmount* | *Görüntülenen ürün fiyatının para birimi tutarı* |
| *product.currencyType* | *Görüntülenen ürün fiyatının para birimi cinsi*  *1: TL*  *2: USD*  *3: EUR* |
| *product.displayPrice* | *Görüntülenen ürün fiyatı (Ürünün indirimler sonucu tanımlanan son fiyat hali)* |
| *product.id* | *Ürünün N11 ID si* |
| *product.productSellerCode* | *Mağaza ürün kodu* |
| *product.title* | *Ürün başlığı* |
| *product.subtitle* | *Ürün alt başlığı* |
| *product.oldPrice* | *Ürüne ait eski baz fiyatı (Güncelleme yapan metodlar sonrasında döner)* |
| *product.price* | *Ürünün baz fiyatı* |
| *product.displayPrice* | *Görüntülenen ürün fiyatı (Ürünün indirimler sonucu tanımlanan son fiyat hali)* |
| *product.saleStatus* | *1: Satış Öncesi(Before\_Sale)*  *2: Satışta(On\_Sale)*  *3: Stok yok (Out of\_Stock)*  *4: Satışa kapalı (Sale\_Closed)* |
| *product.approvalStatus* | *Ürün onay durumu:*  *1: Aktif (Satışta)*  *2: Beklemede*  *3: Yasaklı* |
| *product.stockItems.stockItem.id* | *Ürün stok N11 ID si* |
| *product.stockItems.stockItem.quantity* | *Ürün stok miktarı* |
| *product.stockItems.stockItem.version* | *Ürün stok versiyon numarası* |
| *product.stockItems.stockItem.currencyAmount* | *Ürün stok biriminin para birimi cinsinden tutarı* |
| *product.stockItems.stockItem.displayPrice* | *Ürün stok biriminin görüntülenen ürün fiyatı (Ürünün satıcı indirimi sonucu tanımlanan son fiyat hali)* |
| product.stockItems.stockItem.bundle | *Birlikte satış yapılan ürünler* |
| product.stockItems.stockItem.mpn | *Ürünün üretici parça numarası* |
| product.stockItems.stockItem.gtin | *Ürünün global ticari öğe numarası* |
| *product.stockItems.stockItem.sellerStockCode* | *Ürün stok mağaza kodu* |
| *product.stockItems.stockItem.oldOptionPrice* | *Ürün stok biriminin önceki liste fiyatı (Güncelleme yapan metodlar sonrasında döner)* |
| *product.stockItems.stockItem.optionPrice* | *Ürün stok biriminin liste fiyatı* |
| *product.stockItems.stockItem.attributes.attribute.name* | *Stok tanımının olduğu seçenek adı* |
| *product.stockItems.stockItem.attributes.attribute.value* | *Stok tanımının olduğu seçenek değeri* |

### Ürün Satış Durumu (Product Sale Status)

|  |  |
| --- | --- |
| *product.saleStatus* | *1: Satış Öncesi (Before\_Sale)*  *2: Satışta (On\_Sale)*  *3: Stok yok (Out of\_Stock)*  *4: Satışa kapalı (Sale\_Closed)* |

### Ürün Onay Durumu (Product Approval Status)

|  |  |
| --- | --- |
| *product.approvalStatus* | *Ürün onay durumu:*  *1: Aktif (Satışta)*  *2: Beklemede*  *3: Yasaklı* |

### Ürün İndirim Bilgisi (Product Discount Data)

|  |  |
| --- | --- |
| *discount.discountType* | Mağaza İndirim Tipi:  1: İndirim Tutarı Cinsinden  2: İndirim Oranı Cinsinden  *3: İndirimli Fiyat Cinsinden* |
| *discount.discountValue* | *İndirim Miktarı (indirim tipinde verilen parametreye göre)* |
| *discount.discountStartDate* | *Mağaza indirimi başlama tarihi* |
| *discount.discountEndDate* | *Mağaza indirimi bitiş tarihi* |

### Ürün Durumu (Product Condition)

|  |  |
| --- | --- |
| *product.productCondition* | *Ürün durumu:*  *1: Yeni*  *2: 2. el* |

### Onay Durumu (Approval Status)

|  |  |
| --- | --- |
| productSearch.approvalStatus | Ürünün onay durumu  “Active”: satışda olan ürünler  “Suspended”: Beklemede olan ürünler  “Prohibited”: Yasaklı olan ürünler |

### Ürün Başlığı (Title)

|  |  |
| --- | --- |
| **product.title** | Ürün sayfasında görünecek başlık, en fazla 100 kaakter olmalıdır. |

### Ürün Altbaşlığı (Subtitle)

|  |  |
| --- | --- |
| **product.subtitle** | Ürün sayfasında görünecek altbaşlık, en fazla 100 kaakter olmalıdır. |

### İndirim Tipi (Discount Type)

|  |  |
| --- | --- |
| ***productDiscount.discountType*** | Mağaza İndirim Tipi:  1: İndirim Tutarı Cinsinden  2: İndirim Oranı Cinsinden  *3: İndirimli Fiyat Cinsinden* |

### Sipariş Durumu (SearchData)

|  |  |
| --- | --- |
| *searchData.status* | *Sipariş durumu:*  *“New”: Yeni siparişler*  *“Approved”: Onaylanmış Siparişler*  *“Rejected”: İptal Edilmiş Siparişler*  *“Shipped”: Kargolanmış Siparişler*  *“Delivered”: Teslim Edilen Siparişler*  *“Completed”: Tamamlanmış Siparişler*  *“Claimed”: İptal/İade/Değişim Durumundaki Siparişler* |

### Ödeme Tipi (Payment Type)

|  |  |
| --- | --- |
| *orderList.order.paymentType* | *Ödeme tipi :*  *1: Kredi Kartı*  *2: Bonus Pay*  *3: BKM*  *4: n11 puanı*  *5: Diğer*  *6: Masterpass* |

### Kargo Ödeme Yöntemi (Delivery Fee Type)

|  |  |
| --- | --- |
| *orderList.order.orderItemList.orderItem.deliveryFeeType* | *Kargo ödeme yöntemi:*  *1: N11 Öder*  *2: Alıcı Öder*  *3: Mağaza Öder*  *4: Şartlı Kargo (Alıcı Öder)*  *5: Şartlı Kargo Ücretsiz (Satıcı Öder)* |

### Servis Sipariş Kalemi Tipi (Orderitem Type)

|  |  |
| --- | --- |
| *orderDetail.serviceItemList.serviceItem.orderItemType* | *Sipariş Kalemi Tipi:*  *1: Ürün Sipariş Kalemi*  *2: Servis Sipariş Kalemi*  *3: Gsm Sipariş Kalemi*  *4: Gezi Sipariş Kalemi* |

### Sipariş Durum Bilgisi (Order Status)

|  |  |
| --- | --- |
| *orderList.order.status* | Sipariş durum bilgisi:  1:İşlem Bekliyor  2:İşlemde  3:İptal Edilmiş  4:Geçersiz  5:Tamamlandı |

### Sipariş Maddesi Durum Bilgisi (OrderItem Status)

|  |  |
| --- | --- |
| *orderList.order.orderItemList.orderItem.status* | Sipariş maddesi durum bilgisi:  1:İşlem Bekliyor  2:Ödendi  3:Geçersiz  4:İptal Edilmiş  5:Kabul Edilmiş  6:Kargoda  7:Teslim Edilmiş  8:Reddedilmiş  9:İade Edildi  10:Tamamlandı  11: İptal Talep Edildi  12: İade İptal Değişim Tamamlandı  13:Kargoda İade  14:Kargo Yapılması Gecikmiş,  15:Kabul Edilmiş Ama Zamanında Kargoya Verilmemiş  16:Teslim Edilmiş İade,  *17:Tamamlandıktan Sonra İade* |

### Ödeme Tipi (Payment Type)

|  |  |
| --- | --- |
| *orderDetail.paymentType* | *Ödeme tipi :*  *1: Kredi Kartı*  *2: Bonus Pay*  *3: BKM*  *4: n11 puanı*  *5: Diğer*  *6: Masterpass* |

### Kargo Çoklu Gönderim Durumu (Shipment Allowed)

|  |  |
| --- | --- |
| *shipmentApiModel.combinedShipmentAllowed* | *1’den fazla ürün gönderim durumu true ise 1’den fazla ürün var demek ; false ise tek ürün olduğunu belirtir.* |

prepared by Dogus Planet