****

**API REFERENCE DOCUMENTATION**

n11.com API Enterprise Services, Methods and Usage Examples

|  |  |
| --- | --- |
| **Change Log** | |
| 20.05.2013 | Document first version was created |
| 24.05.2013 | <status> tag content has been changed, new definitions were added to the data templates, the domain names on the ShipmentService response side were updated. |
| 07.06.2013 | "sellerInvoiceAmount" field under the <itemlist> item tag, and taxId, taxOffice and e-mail fields have been added to the order detail service. |
| 26.06.2013 | Product attributes field added to the SaveProduct service. |
| 30.07.2013 | ProductSaleStatus statuses were revised. |
| 17.12.2013 | OrderItemReject, OrderItemAccept, and MakeOrderItemShipment were arranged to enable making transactions for multiple order items. |
| 21.05.2014 | stockItems.stockItem.optionPrice field has been added. |
| 30.05.2014 | All services were updated. |
| 08.08.2014 | SpecialProductInfoList field added to SaveProduct service |
| 22.08.2014 | Sample Java Request Codes added |
| 10.03.2015 | ProductAnswer Service Added |
| 30.09.2015 | Conditional shipping option added to Delivery Fee Type field |
| 30.09.2015 | Category Service Authentication added |
| 07.11.2016 | Definitions and sample requests added for GetCategoryAttributesId, GetCategoryAttributeValue methods. |
| 17.11.2016 | Masterpass added to payment types. |
| 26.12.2016 | Bundle, Gtin, Mpn fields added to the SaveProduct service. |
| 03.03.2017 | GetNeighborhoods method added, the Neighborhood area added to the DetailedOrderList and OrderDetail fields. "CLAIMED" added to the searchData.status field in order services. |
| 04.07.2017 | SettlementService methods added. |
| 20.02.2018 | SortForUpdate added to DetailedOrderList and OrderList's OrderDataListRequest, and its type is determined as Boolean (true/false). OEM (Part/Product connection number) added to the inventory information for product shipping and listing. Order item status information updated as “11.Return/Cancellation/ Change Requested”. |
| 17.12.2019 | Filtering of products with delayed shipment status in DetailedOrderList and OrderList |
| 09.04.2019 | A new endpoint named ProductAprovalAllStatusRequest added to ProductService. With this endpoint, product-status number information is obtained. |
| 02.07.2019 | Domestic field added to the SaveProduct service. It is marked whether the product added/updated with this field is domestic production. |
| 10.07.2019 | isDomestic field has been added to GetProductList service, which keeps whether the product is domestic or not. The domestic area, which retrieves domestic production information, has been added to the GetProductByProductId and GetProductBySellerCode services. |
| 17.09.2019 | PaymentType information from orderDetail and detailedOrderList methods, where order information is taken, has been updated. |
| 25.10.2019 | TicketService document information has been added. orderList.order. paymentType,orderList.order.orderItemList.orderItem.status, product.title and product.subtitle fields have been updated. |
| 15.01.2020 | Sap Commission E-Invoice Detail Service document information added. |
| 16.01.2020 | Information of the TicketService, newly added ticketReason and ticketRead services have been added to the document. |
| 18.01.2020 | Document information of Sap Bank Statement E-Invoice Service has been added. |
| 03.02.2020 | TicketListingAssignedToSellerSARequest arranged as -> TicketListingAssignedToSellerRequest |
| 14.02.2020 | ClaimCancelService document information has been added. |
| 24.02.2020 | Product grouping information has been added to GetProductByProductId, GetProductBySellerCode and SaveProduct requests. (groupAttribute, groupItemCode, itemName) |
| 11.03.2020 | stockItems.stockItem.n11CatalogId field added. |
| 11.03.2020 | numberOfPackages field added to the OrderItemAccept service. |
| 27.03.2020 | ClaimExchangeService document details added. |

17.12.2018

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

Contents

[1. INTRODUCTION 5](#_Toc34123738)

[2. VERIFICATION AND SECURITY MECHANISM 6](#_Toc34123739)

[2.1 Security Mechanism 6](#_Toc34123740)

[3. METHOD INFORMATION 8](#_Toc34123741)

[3.1 Category Services (CategoryService) 8](#_Toc34123742)

[Showing Category Attribute and Value (GetCategoryAttributes) 8](#_Toc34123743)

[Showing Category Attribute (GetCategoryAttributesId) 10](#_Toc34123744)

[Showing Category Attribute (GetCategoryAttributeValue) 12](#_Toc34123745)

[Showing Parent Category Information (GetParentCategory) 14](#_Toc34123746)

[Listing Sub-Category Information (GetSubCategories) 16](#_Toc34123747)

[Top Category Information (GetTopLevelCategories) 17](#_Toc34123748)

[3.2 City Services (CityService) 19](#_Toc34123749)

[Listing The Cities (GetCities) 19](#_Toc34123750)

[Showing City Information (GetCity) 20](#_Toc34123751)

[Display of District Information (GetDistrict) 21](#_Toc34123752)

[Displaying Neighborhood Information (GetNeighborhoods) 23](#_Toc34123753)

[3.3 Product Services (ProductService) 25](#_Toc34123754)

[Getting Product Information with N11 Product ID (GetProductByProductId) 25](#_Toc34123755)

[Getting Product Information with Store Product Code (GetProductBySellerCode) 28](#_Toc34123756)

[Product Listing (GetProductList) 32](#_Toc34123757)

[Save Product (Save Product) 34](#_Toc34123758)

[Product Search (SearchProducts) 40](#_Toc34123759)

[Product Deletion with N11 Product ID (DeleteProductById) 42](#_Toc34123760)

[Product Deletion with Store Product Code (DeleteProductBySellerCode) 44](#_Toc34123761)

[Update Discount Amount with N11 Product ID (UpdateDiscountValueByProductId) 46](#_Toc34123762)

[Update Discount Amount with Store Product Code (UpdateDiscountValueBySellerCode) 48](#_Toc34123763)

[Product Price Update with N11 Product Code (UpdateProductPriceById) 50](#_Toc34123764)

[Update Product Price with Store Product Code (UpdateProductPriceBySellerCode) 52](#_Toc34123765)

[Updating Basic Product Attributes (UpdateProductBasic) 54](#_Toc34123766)

[Listing Product Questions (GetProductQuestionList) 57](#_Toc34123767)

[Detailing Product Questions (GetProductQuestionDetail) 59](#_Toc34123768)

[Answering Product Questions (SaveProductAnswer) 61](#_Toc34123769)

[Get the Product-Status Number (ProductAllStatusCountsRequest) 62](#_Toc34123770)

[3.4 Product Sales Services (ProductSellingService) 64](#_Toc34123771)

[Start Selling the Product with N11 Product ID (StartSellingProductByProductId) 64](#_Toc34123772)

[Start Selling the Product with the Store Product Code (StartSellingProductBySellerCode) 66](#_Toc34123773)

[Stop Selling the Product with N11 Product ID (StopSellingProductByProductId) 67](#_Toc34123774)

[Stop Selling the Product with the Store Product Code (StopSellingProductBySellerCode) 69](#_Toc34123775)

[3.5 Product Stock Services (ProductStockService) 70](#_Toc34123776)

[Listing Stock Information with N11 Product ID (GetProductStockByProductId) 70](#_Toc34123777)

[Listing Stock Information with Store Product Code (GetProductStockBySellerCode) 72](#_Toc34123778)

[Updating Stock Quantity with Stock Attributes (DeleteAndUpdateStockByStockAttributes ) 74](#_Toc34123779)

[Stock Update with N11 Product Stock ID (UpdateStockByStockId) 76](#_Toc34123780)

[Stock Update with Store Stock Code (UpdateStockByStockSellerCode) 79](#_Toc34123781)

[Increasing Stock Quantity with Stock Attributes (IncreaseStockByStockAttributes) 81](#_Toc34123782)

[Increasing Stock Quantity with N11 Product Stock ID (IncreaseStockByStockId) 83](#_Toc34123783)

[Increasing Stock Quantity with Store Stock Code (IncreaseStockByStockSellerCode) 85](#_Toc34123784)

[3.6 Order Services (OrderService) 87](#_Toc34123785)

[Detailed Order Listing (DetailedOrderList) 87](#_Toc34123786)

[Order Listing (OrderList) 91](#_Toc34123787)

[Order Detail (OrderDetail) 94](#_Toc34123788)

[Acceptance of Order Item (OrderItemAccept) 98](#_Toc34123789)

[Rejection of Order Item (OrderItemReject) 101](#_Toc34123790)

[Shipment of the Order Item (MakeOrderItemShipment) 104](#_Toc34123791)

[3.7 Shipping Company Service (ShipmentCompanyService) 107](#_Toc34123792)

[Listing the Shipping Companies (GetShipmentCompanies) 107](#_Toc34123793)

[3.8 Shipment Template Service (ShipmentService) 109](#_Toc34123794)

[Shipment Template Listing (GetShipmentTemplate) 109](#_Toc34123795)

[Creating and Updating Shipment Templates (CreateOrUpdateShipmentTemplate) 112](#_Toc34123796)

[Shipment Template Listing (GetShipmentTemplateList) 117](#_Toc34123797)

[3.9 Sales/Settlement Service (SettlementService) 121](#_Toc34123798)

[Sales/Settlement Listing (GetSettlementList) 121](#_Toc34123799)

[Sales/Settlement Detailing (GetSettlementDetail) 123](#_Toc34123800)

[3.10 Store Request Service (TicketService) 127](#_Toc34123801)

[Listing the Requests Assigned to the Store (TicketListingAssignedToSeller) 127](#_Toc34123802)

[Listing the Requests Opened by the Store (TicketListingBelongsToSeller) 132](#_Toc34123803)

[Store's Response to Requests (TicketAnswer) 136](#_Toc34123804)

[Request Opening of the Store (TicketCreate) 137](#_Toc34123805)

[Store Request Reasons (TicketReasons) 138](#_Toc34123806)

[Request and Response Reading (TicketRead) 144](#_Toc34123807)

[3.11 Sap Commission E-Invoice Detail Service (SapCommissionEInvoiceDetailService) 145](#_Toc34123808)

[Invoice Detailing (GetSapCommissionEInvoiceDetail) 145](#_Toc34123809)

[3.12 Sap Bank Statement E-Invoice Service (SapBankStatementEInvoiceService) 147](#_Toc34123810)

[Bank Statement (GetSapBankStatementEInvoice) 148](#_Toc34123811)

[3.13 Order Cancellation Requests Service (ClaimCancelService) 150](#_Toc34123812)

[Listing the Cancellation Requests (ClaimCancelList) 150](#_Toc34123813)

[Rejecting the Cancellation Request (ClaimCancelDeny) 153](#_Toc34123814)

[Listing the Reasons for Cancellation Request Rejection (ClaimCancelDenyReasonType) 154](#_Toc34123815)

[Confirming the Cancellation Request (ClaimCancelApprove) 156](#_Toc34123816)

[3.14 Commonly Used Data Templates 157](#_Toc34123817)

[Authorization (Authentication) 157](#_Toc34123818)

[Result Status Information (Result) 157](#_Toc34123819)

[Pagination 157](#_Toc34123820)

[Product Summary Information (Product Summary Data) 157](#_Toc34123821)

[Product Sale Status (Product Sale Status) 158](#_Toc34123822)

[Product Approval Status (Product Approval Status) 158](#_Toc34123823)

[Product Discount Information (Product Discount Data) 158](#_Toc34123824)

[Product Status (Product Condition) 159](#_Toc34123825)

[Approval Status (Approval Status) 159](#_Toc34123826)

[Product Title (Title) 159](#_Toc34123827)

[Product Subtitle (Subtitle) 159](#_Toc34123828)

[Discount Type (Discount Type) 159](#_Toc34123829)

[Order Status (SearchData) 160](#_Toc34123830)

[Payment Type (Payment Type) 160](#_Toc34123831)

[Shipment Payment Method (Delivery Fee Type) 160](#_Toc34123832)

[Service Order Item Type (Orderitem Type) 160](#_Toc34123833)

[Order Status Information (Order Status) 160](#_Toc34123834)

[Order Item Status Information (OrderItem Status) 161](#_Toc34123835)

[Payment Type (Payment Type) 161](#_Toc34123836)

[Shipment Multiple Shipment Status (Shipment Allowed) 162](#_Toc34123837)

[Order Cancellation Request Status (Claim Cancel Status) 162](#_Toc34123838)

[Transaction Executioner (Claim Executer Type) 162](#_Toc34123839)

[Cancellation Request Detail (Claim Search Type) 162](#_Toc34123840)

[Cancellation Request Search Date Type (Claim Cancel Date Type) 163](#_Toc34123841)

# 1. INTRODUCTION

This document contains API integration information that covers many of the actions that n11 users can do through the site or the Store Management Center. The services are SOAP-based web services.

The addresses of the web service definition files are listed below.

<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>

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

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

[*https://api.n11.com/ws/SapCommissionEInvoiceDetailService.wsdl*](https://api.n11.com/ws/SapCommissionEInvoiceDetailService.wsdl)

[*https://api.n11.com/ws/SapBankStatementEInvoicelService.wsdl*](https://api.n11.com/ws/SapBankStatementEInvoicelService.wsdl)

[*https://api.n11.com/ws /ClaimCancelService.wsdl*](https://api.n11.com/ws%20/ClaimCancelService.wsdl)

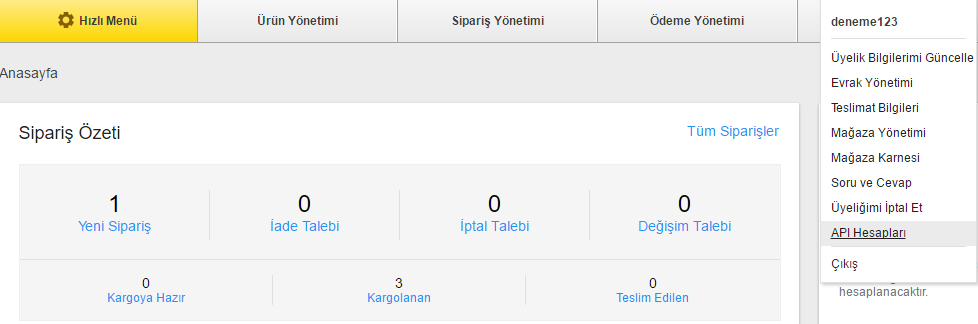
# 2. VERIFICATION AND SECURITY MECHANISM

## 2.1 Security Mechanism

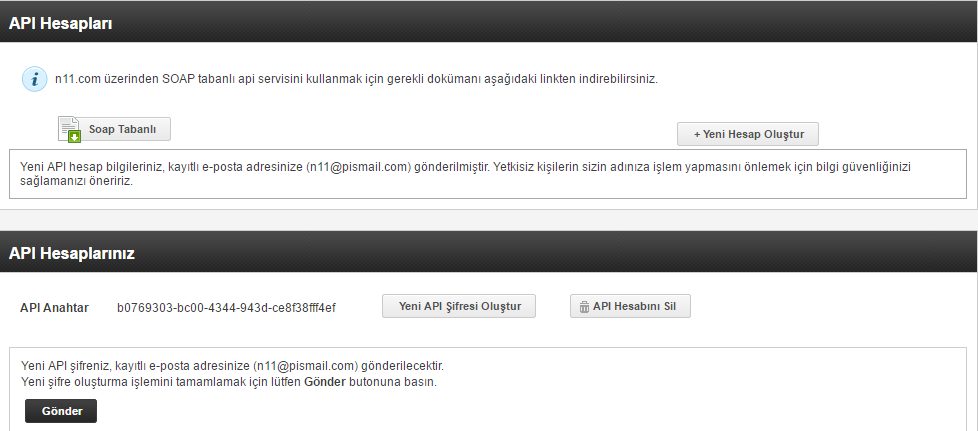
As a security mechanism;

* Authentication: After logging into the Store Management Center, it is provided by sending the security key **(API Key)**  taken from My Account - API Accounts section under the main menu in every request. The API password is sent to user’s e-mail address. It is mandatory to send the API Key and API Password. The identity of the store is determined with these two parameters. In this way, it is ensured that the store only inquires about its products and other information.

1) API key and password / My Account - API Accounts menu



2) API account is created with the "Create New Account" function under API Accounts. The password of the created account is sent by e-mail. If desired, API password can be re-created or API account can be deleted.



3) The obtained API Key and API password are sent when requested, verifying that the authorized user calls the relevant service and the service calls are successfully terminated.

# 3. METHOD INFORMATION

## 3.1 Category Services (CategoryService)

WSDL address for category services:

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

### Showing Category Attribute and Value (GetCategoryAttributes)

*The desired category can be from the top level category or other level categories, it is the method used to list the attributes and values of these categories.*

**GetCategoryAttributesRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| **categoryId** | Category ID whose attributes are to be listed |
| pagingData | [See: Pagination](#_Sayfalandırma_(Pagination)) |

**\*Bold fields are mandatory fields.**

**GetCategoryAttributesResponse**

|  |  |
| --- | --- |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| pagingData | [See: Pagination](#_Sayfalandırma_(Pagination)_1) |
| category.attributeList.attribute.id | Id information given to the attribute in our system |
| category.attributeList.attribute.mandatory | Information about the obligation to enter each attribute is learned through this field. |
| category.attributeList.attribute.multipleSelect | Whether an attribute can have more than one value is learned from this field. |
| category.attributeList.attribute.name | Contains the attribute name information |
| category.attributeList.attribute.priority | Category Priority |
| category.attributeList.attribute.valueList.value.id | Contains the value code to be entered into attribute |
| category.attributeList.attribute.valueList.value.name | Contains the name information to be entered into attribute |

**GetCategoryAttributes Sample Call**

**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>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

</pagingData> </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>

<metadata>

<currentPage>1</currentPage>

<pageSize>100</pageSize>

<totalCount>2695</totalCount>

<pageCount>27</pageCount>

</metadata> <attributeList>

<attribute>

<id>354189900</id>

<mandatory>true</mandatory>

<multipleSelect>false</multipleSelect>

<name>Marka</name>

<priority>4.0</priority>

<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 Sample Call (Java)**

public static void main(String[] args) {  
  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";

int currentPageVal = 0;

int pageSizeVal = 100;  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 GetCategoryAttributesRequest request = new GetCategoryAttributesRequest();  
 request.setAuth(authentication);  
 request.setCategoryId(categoryIdValue);

RequestPagingData pagingData = new RequestPagingData();

pagingData.setCurrentPage(currentPageVal);

pagingData.setPageSize(pageSizeVal);

request.setPagingData(pagingData);

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() + ")");  
 }  
  
 }}

### Showing Category Attribute (GetCategoryAttributesId)

*The desired category can be from the top level category or other level categories, it is the method used to list the attributes belonging to these categories.*

**GetCategoryAttributesIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| **categoryId** | Category ID whose attributes are to be listed |

**\*Bold fields are mandatory fields.**

**GetCategoryAttributesIdResponse**

|  |  |
| --- | --- |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| category.attributeList.attribute.id | Id information given to the attribute in our system |
| category.attributeList.attribute.mandatory | Information about the obligation to enter each attribute is learned through this field. |
| category.attributeList.attribute.multipleSelect | Whether an attribute can have more than one value is learned from this field. |
| category.attributeList.attribute.name | Contains the attribute name information |

**GetCategoryAttributesId Sample Call**

**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);

RequestPagingData pagingData = new RequestPagingData();

pagingData.setCurrentPage(currentPageVal);

pagingData.setPageSize(pageSizeVal);

request.setPagingData(pagingData);

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());

}

*}*

### Showing Category Attribute (GetCategoryAttributeValue)

By entering the id information (category.attributeList.attribute.id) given to the attribute in our system, it lists the values of that attribute.

**GetCategoryAttributeValueRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| **categoryProductAttributeId** | The id information given to the product attribute in our system |
| pagingData | [See: Pagination](#_Sayfalandırma_(Pagination)) |

**\*Bold fields are mandatory fields.**

**GetCategoryAttributeValueResponse**

|  |  |
| --- | --- |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| categoryProductAttributeValueList. categoryProductAttributeValue.id | The ID value given in our system to the value information of an attribute |
| categoryProductAttributeValueList. categoryProductAttributeValue.name | Name information of the value of an attribute |
| pagingData | [See: Pagination](#_Sayfalandırma_(Pagination)_1) |

**GetCategoryAttributeValue Sample Call**

**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>

<pagingData>

<currentPage>1</currentPage>

<pageSize>100</pageSize>

<totalCount>7</totalCount>

<pageCount>1</pageCount>

</pagingData>

</ns3:GetCategoryAttributeValueResponse>

</env:Body>

</env:Envelope>

**GetCategoryAttributeValue Sample Call (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());

} }

### Showing Parent Category Information (GetParentCategory)

This method should be used to reach the first level parent categories of the category whose code is given. To reach second level parent categories (E.g. In the "Leather shoes -> Shoes -> Clothes" category tree, "Clothes" information) service should be called again by giving the code of the first level parent category (Ex.Shoes).

If the inquired category cannot be found in the system, a 'category not found' error is received. If the related category does not have any parent category, the “parentCategory” information is not included in the response.

**GetParentCategoryRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| **categoryId** | Subcategory id information |

**\*Bold fields are mandatory fields.**

**GetParentCategoryResponse**

|  |  |
| --- | --- |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| category.id | Subcategory id information |
| category.name | Category Name |
| category.parentCategory.id | Parent category id information |
| category.parentCategory.name | Parent category name |

**GetParentCategory Sample Call**

**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:GetParentCategoryResponse 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:GetParentCategoryResponse>

</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());  
}

### Listing Sub-Category Information (GetSubCategories)

This method should be used to reach the first level sub-categories of the category whose code is given. To reach second level sub-categories (E.g. "Leather shoes" in the "Dress -> Shoes -> Leather shoes" category tree) service should be called again by giving the code of the first level parent category (Ex.Shoes).

If the inquired category cannot be found in the system, an error is received. If the related category does not have any sub-categories, the response information returns empty.

**GetSubCategoriesRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| **categoryId** | Subcategory id information |

**\*Bold fields are mandatory fields**

**GetParentCategoryResponse**

|  |  |
| --- | --- |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| category.id | Subcategory id information |
| category.name | Category Name |
| subCategoryList.subCategory.id | Sub-category id information |
| subCategoryList.subCategory.name | Sub-category name |

**GetSubCategory Sample Call**

**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()); }}

### Top Category Information (GetTopLevelCategories)

This method is used to see all top level categories defined on n11. In response, n11 category code and category name are returned.

**GetTopLevelCategoriesRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |

**\*Bold fields are mandatory fields.**

**GetTopLevelCategoriesResponse**

|  |  |
| --- | --- |
| categories.category.id | Code of the category on the n11 side |
| categories.category.name | Category name |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |

**GetTopLevelCategories Sample Call**

**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 Sample Call (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 City Services (CityService)

WSDL address for city services: <https://api.n11.com/ws/CityService.wsdl>

### Listing The Cities (GetCities)

This method is used to list the cities registered with the codes and license plate numbers. When it is desired to take action on the address, the city codes obtained through this service are used.

Since it is a public service, no security check is performed during the use of the service.

**GetCitiesResponse**

|  |  |
| --- | --- |
| Result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| cities.city.cityCode | License code of the city |
| cities.city.cityId | Code of the city on the n11 side |
| cities.city.cityName | City Name |

**GetCities Sample Call**

**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 Sample Call (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() + ")");

}

}

### Showing City Information (GetCity)

This method should be used to find out the city code of the city with the city license plate number and which city it is in.

Since it is a public service, no security check is performed during the use of the service.

If the inquired city cannot be found in the system, a 'city not found' error is received.

**GetCityRequest**

|  |  |
| --- | --- |
| **cityCode** | License code of the city |

**\*Bold fields are mandatory fields.**

**GetCityResponse**

|  |  |
| --- | --- |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| city.cityCode | City code |
| city.cityId | Id given to the city by n11 |
| city.cityName | City name |

**GetCity Sample Call**

**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 Sample Call (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() + ")" );  
}

### Display of District Information (GetDistrict)

This method should be used to list the districts of the city where the license plate code is given. District code is used in address adding/updating processes.

Since it is a public service, no security check is performed during the use of the service.

If the inquired city cannot be found in the system, a 'city not found' error is received.

**GetDistrictRequest**

|  |  |
| --- | --- |
| **cityCode** | License code of the city |

**\*Bold fields are mandatory fields.**

**GetDistrictResponse**

|  |  |
| --- | --- |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| districts.district.id | License code of the city |
| districts.district.name | Code of the district on the n11 side |

**GetDistrict Sample Call**

**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 Sample Call (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());  
 }  
}

### Displaying Neighborhood Information (GetNeighborhoods)

This method should be used to list the neighborhoods whose district code is provided.

Since it is a public service, no security check is performed during the use of the service.

If the inquired district cannot be found in the system, a 'district not found' error is received.

**GetNeighborhoodsRequest**

|  |  |
| --- | --- |
| **districtId** | District Code |

**\*Bold fields are mandatory fields.**

**GetNeighborhoodsResponse**

|  |  |
| --- | --- |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| Neighborhoods.neighborhood.id | Neighborhood code |
| Neighborhoods.neighborhood.name | Code of the neighborhood on the n11 side |

**GetNeighborhoods Sample Call**

**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 Sample Call (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 Product Services (ProductService)

WSDL address for the product:

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

### Getting Product Information with N11 Product ID (GetProductByProductId)

Retrieves the information of the product registered in the system using the N11 product ID.

**GetProductByIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Product N11 ID* |

**\*Bold fields are mandatory fields.**

**GetProductByIdResponse**

|  |  |
| --- | --- |
| result | [See: Result](#_Sonuç_Durum_Bilgisi(Result)) |
| *product.currencyAmount* | *Currency amount of the displayed product price* |
| *product.currencyType* | *Currency of the displayed product price (USD/EUR/TRY)* |
| *product.displayPrice* | *Product price displayed (The final price of the product as a result of discounts)* |
| *product.id* | *Product N11 ID* |
| *product.price* | *List price of the product* |
| *product.productSellerCode* | *Store product code* |
| *product.saleStatus* | [*See: Product Sales Status*](#_Ürün_Satış_Durumu) |
| *product.approvalStatus* | [*See: Product Approval Status*](#_Ürün_Onay_Durumu) |
| *product.subtitle* | *Product subtitle* |
| *product.title* | *Product title* |
| *product.desctiption* | *Product Description* |
| *product.domestic* | *Indicates whether the product is domestic production. Gets true / false as a Boolean.* |
| *product.category.fullName* | *Product category information with all category subtitles* |
| *product.category.id* | *Product category id* |
| *product.category.name* | *Product category name* |
| *product.images.image.order* | *Image priority number* |
| *product.images.image.url* | *Image url address* |
| *product.preparingDay* | *Delivery time of the product (in days)* |
| *product.productCondition* | [*See: Product Status*](#_Ürün_Durumu_(Product) |
| *product.saleEndDate* | *Last selling date of the product* |
| *product.saleStartDate* | *Date the product was launched* |
| *product.shipmentTemplate* | *Shipment template set for the product* |
| *product.discount* | [*See: Product Discount Information*](#_Ürün_İndirim_Bilgisi_1) |
| *product.stockItems.stockItem.id* | *Product stock N11 ID* |
| *product.stockItems.stockItem.quantity* | *Product stock quantity* |
| *product.stockItems.stockItem.version* | *Product stock version number* |
| *product.stockItems.stockItem.currencyAmount* | *Amount of product stock unit in currency* |
| *product.stockItems.stockItem.displayPrice* | *The displayed product price of the product stock unit (the final price defined as a result of the seller discount of the product)* |
| *product.stockItems.stockItem.optionPrice* | *List price of the product stock unit* |
| *product.stockItems.stockItem.sellerStockCode* | *Product stock store code* |
| product.stockItems.stockItem.bundle | *Products that are sold together* |
| product.stockItems.stockItem.mpn | *Manufacturer part number of the product* |
| product.stockItems.stockItem.gtin | *Global trade item number of the product* |
| product.stockItems.stockItem.oem | *Part/Product connection number* |
| *product.stockItems.stockItem.attributes.attribute.name* | *Option name with stock definition* |
| *product.stockItems.stockItem.attributes.attribute.value* | *Option value with stock definition* |
| *product.groupAttribute* | *Attribute name where the product is grouped* |
| *product.groupItemCode* | *Name of the attribute group where the product is grouped (the seller gives it)* |
| *product.itemName* | *Attribute value where the product is grouped* |

**GetProductByProductId Sample Call**

**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>

<oem></oem>

<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>

<oem></oem>

<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>

<domestic>true</domestic>

<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/>

<groupAttribute/>

<groupItemCode/>

<itemName/>

</product>

</ns3:GetProductByProductIdResponse>

</env:Body>

</env:Envelope>

**GetProductByProductId Sample Call (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());  
}

### Getting Product Information with Store Product Code (GetProductBySellerCode)

Retrieves the information of the product registered in the system using the store product code.

**GetProductBySellerCodeRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***sellerCode*** | *Product store code* |

**\*Bold fields are mandatory fields.**

**GetProductBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product.currencyAmount* | *Currency amount of the displayed product price* |
| *product.currencyType* | *Currency of the displayed product price (USD/EUR/TRY)* |
| *product.displayPrice* | *Product price displayed (The final price of the product as a result of discounts)* |
| *product.id* | *Product N11 ID* |
| *product.price* | *List price of the product* |
| *product.productSellerCode* | *Store product code* |
| *product.saleStatus* | [*See: Product Sales Status*](#_Ürün_Satış_Durumu_1) |
| *Product.approvalStatus* | [*See: Product Approval Status*](#_Ürün_Onay_Durumu_1) |
| *product.subtitle* | *Product subtitle* |
| *product.title* | *Product title* |
| *product.domestic* | *Indicates whether the product is domestic production. Gets true / false as a Boolean.* |
| *product.category.fullName* | *Product category information with all category subtitles* |
| *product.category.id* | *Product category id* |
| *product.category.name* | *Product category name* |
| *product.images.image.order* | *Image priority number* |
| *product.images.image.url* | *Image url address* |
| *product.preparingDay* | *Delivery time of the product (in days)* |
| *product.productCondition* | [*See: Product Status*](#_Ürün_Durumu_(Product) |
| *product.saleEndDate* | *Last selling date of the product* |
| *product.saleStartDate* | *Date the product was launched* |
| *product.shipmentTemplate* | *Shipment template set for the product* |
| *product.discount* | [*See: Product Discount Information*](#_Ürün_İndirim_Bilgisi_1) |
| *product.stockItems.stockItem.id* | *Product stock N11 ID* |
| *product.stockItems.stockItem.quantity* | *Product stock quantity* |
| *product.stockItems.stockItem.version* | *Product stock version number* |
| *product.stockItems.stockItem.currencyAmount* | *Amount of product stock unit in currency* |
| *product.stockItems.stockItem.displayPrice* | *The displayed product price of the product stock unit (the final price defined as a result of the seller discount of the product)* |
| *product.stockItems.stockItem.optionPrice* | *List price of the product stock unit* |
| *product.stockItems.stockItem.sellerStockCode* | *Product stock store code* |
| product.stockItems.stockItem.bundle | *Products that are sold together* |
| product.stockItems.stockItem.mpn | *Manufacturer part number of the product* |
| product.stockItems.stockItem.gtin | *Global trade item number of the product* |
| product.stockItems.stockItem.oem | *Part/Product connection number* |
| *product.stockItems.stockItem.attributes.attribute.name* | *Option name with stock definition* |
| *product.stockItems.stockItem.attributes.attribute.value* | *Option value with stock definition* |
| *product.groupAttribute* | *Attribute name where the product is grouped* |
| *product.groupItemCode* | *Name of the attribute group where the product is grouped (the seller gives it)* |
| *product.itemName* | *Attribute value where the product is grouped* |

**GetProductBySellerCode Sample Call**

**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>

<oem></oem>

<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>

<oem></oem>

<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>

<domestic>true</domestic>

<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/>

<groupAttribute/>

<groupItemCode/>

<itemName/>

</product>

</ns3:GetProductBySellerCodeResponse>

</env:Body>

</env:Envelope>

**GetProductByProductSellerCode Sample Call (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);  
}

### Product Listing (GetProductList)

It is used to list products in the system. The method brings summary information about the product. The detailed information of the product can be accessed using the **"GetProductBySellerCode"** method using the **"productSellerCode"** information retrieved from this method or the **"GetProductById"** method using the **"id"** information.

Results are retrieved with the paging information. The number of products to be listed on each page with the "**itemsPerPage"** parameter, and the number of the page to be listed with the **"currentPage"** parameter can be checked. If no value is entered, **100** products are listed on each page and the first page is listed.

Total number of products, number of pages, current page and number of products per page are given to users in **“pagingData”**.

**GetProductListRequest**

|  |  |
| --- | --- |
| **auth** | [See: Authorization](#_Yetkilendirme_(Authentication)) |
| pagingData | [See: Pagination](#_Sayfalandırma_(Pagination)) |

**\*Bold fields are mandatory fields.**

**GetProductListResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**GetProductList Sample Call**

**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>

<isDomestic>true</isDomestic>

<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>

<oem></oem>

<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>

<oem></oem>

<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 Sample Call (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());  
 }  
}

### Save Product (Save Product)

It is used to create a new product or update an existing product. If the data with the product code named as **"productSellerCode"** shows a registered product for this store, this product is updated, otherwise a new product is created.

In the **“stockItems”**, stock entries of both products (such as number, size) and products without options can be entered. For products without options, the "**attributes**" field can be left blank.

“**Category.id**” should include the category N11 ID information in which the product will be included, the category list can start with **CategoryService GetParentCategory** and get it with the **GetSubCategories** methods, the category in which the product will be included should be at the lowest level.

In the "**templateName**", a shipment template defined for this store should be used in the system. To define shipment templates, it can be created manually by following the My Account> Shipment Information menu steps from the Sales Office screens or **createOrUpdateShipmentTemplate** method can be used from within **ShipmentService**. SpecialProductInfoList is the field where you can enter the special description section for the travel category products.

**SaveProductRequest**

|  |  |  |
| --- | --- | --- |
| Auth | [See: Authorization](#_Yetkilendirme_(Authentication)) | |
| **product.productSellerCode** | Product Store Code | |
| **product.title** | [See: title](#_Ürün_Başlığı_(Title)) | |
| **product.subtitle** | [See: subtitle](#_Ürün_Altbaşlığı_(Subtitle)) | |
| **product.description** | Product description information (can be HTML) | |
| product.attributes.attribute | The field where name and value of the product attributes are entered | |
| **product.category.id** | Product category number | |
| **product.price** | Product base price | |
| **product.currencyType** | Product list price currency | |
| **product.images.image.url** | Product image URL address | |
| **product.images.image.order** | Order of product display | |
| product.saleStartDate | Product sales start date (dd/MM/yyyy), the product will be sold on the same day if shipped blank | |
| product.saleEndDate | Product selling end date (dd/MM/yyyy) is assigned to a very advanced date if it is left blank | |
| product.productionDate | Product production date (dd/MM/yyyy) | |
| product.expirationDate | Product expiration date (dd/MM/yyyy) | |
| **product.productCondition** | [See: Product Status](#_Ürün_Durumu_(Product_1) | |
| **product.preparingDay** | Product delivery time (in days) | |
| product. domestic | Indicates whether the product is domestic production. Gets true / false as a Boolean. | |
| product.discount | [See: Product Discount Information](#_Ürün_İndirim_Bilgisi_2) | |
| **product.shipmentTemplate** | Shipment Template Name | |
| **product.stockItems.stockItem.quantity** | The amount of stock | |
| product.stockItems.stockItem.sellerStockCode | Store product stock code | |
| product.stockItems.stockItem.attributes.attribute.name | If the product has options, this information is entered, you can use a value you specify as a name. | |
| product.stockItems.stockItem.attributes.attribute.value | If the product has options, this information is entered, you can use a value you set as the value. | |
| product.stockItems.stockItem.optionPrice | List price of the product stock unit | |
| product.stockItems.stockItem.bundle | *Products that are sold together* | |
| product.stockItems.stockItem.mpn | *Manufacturer part number of the product* | |
| product.stockItems.stockItem.gtin | *Global trade item number of the product* | |
| product.stockItems.stockItem.oem | *Part/Product connection number* | |
| specialProductInfoList.specialProductInf.key | The field where travel category product description information is entered. To This Area; An explanation is created by entering Tour Program, CancelReturn Requirements, Additional Services information. | |
| specialProductInfoList.specialProductInf.value | Html data corresponding to Key | |
| *product.groupAttribute* | *Attribute name where the product is grouped* |
| *product.groupItemCode* | *Name of the attribute group where the product is grouped (the seller gives it)* |
| *product.itemName* | *Attribute value where the product is grouped* |
| *product.maxPurchaseQuantity* | *Maximum number of purchased products* |

**\*Bold fields are mandatory fields.**

**SaveProductResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**SaveProduct Sample Call**

**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>

<maxPurchaseQuantity>500</maxPurchaseQuantity>

<title>Lorem Ipsum</title>

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

<description>Hello World!</description>

<category>

<id>1000476</id>

</category>

<specialProductInfoList/>

<price>3000</price>

<domestic>true</domestic>

<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>

<oem></oem>

<n11CatalogId></n11CatalogId>

<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>

<oem></oem>

<n11CatalogId></n11CatalogId>

<quantity>5</quantity>

<sellerStockCode>StokKodu1234567890-2</sellerStockCode>

<attributes>

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

<attribute>

<name>Renk</name>

<value>Kırmızı</value>

</attribute>

</attributes>

<optionPrice/>

</stockItem>

</stockItems>

<groupAttribute></groupAttribute>

<groupItemCode></groupItemCode>

<itemName></itemName>

</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>

<isDomestic>true</isDomestic>

<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/>

<oem/>

<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/>

<oem/>

<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 Sample Call (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());

}

### Product Search (SearchProducts)

Used to search for store products.

**SearchProductsRequest**

|  |  |
| --- | --- |
| **Auth** | [See: Authorization](#_Yetkilendirme_(Authentication)) |
| pagingData.currentPage | Requested page after searching |
| pagingData.pageSize | Number of items to display on one page |
| productSearch.name | Word to be searched |
| productSearch.saleDate.startDate | The date product was launched |
| productSearch.saleDate.endDate | The date product’s sale ends |
| approvalStatus | [See: ApprovalStatus](#_Onay_Durumu_(Approval) |

**\*Bold fields are mandatory fields.**

**SearchProductsResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**SearchProducts Sample Call**

**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 Sample Call (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());

}

}

### Product Deletion with N11 Product ID (DeleteProductById)

Used to delete a registered product using N11 Id.

**DeleteProductBuIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Product N11 ID information* |

**\*Bold fields are mandatory fields.**

**DeleteProductByIdResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**DeleteProductById Sample Call**

**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 Sample Call (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); }

### Product Deletion with Store Product Code (DeleteProductBySellerCode)

Used to delete a registered product using the store product code.

**DeleteProductBySellerCodeRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Product code of the product on the store* |

**\*Bold fields are mandatory fields.**

**DeleteProductBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**DeleteProductBySellerCode Sample Call**

**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 Sample Call (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.");

}

### Update Discount Amount with N11 Product ID (UpdateDiscountValueByProductId)

It is used to update discount information using the N11 product ID of a product. The amount of discount entered is applied to the product listing price. As well as the difference between the list price and the discount price of the product, a discount is applied to the product stock unit prices.

**UpdateDiscountValueByProductIdRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *N11 product code of the product* |
| ***discountType*** | [*See: DiscountType*](#_İndirim_Tipi_(Discount) |
| ***productDiscount.discountValue*** | *Discount amount (according to the parameter given in the discount type)* |
| *productDiscount.discountStartDate* | *Store discount start date* |
| *productDiscount.discountEndDate* | *Store discount end date* |

**\*Bold fields are mandatory fields.**

**UpdateDiscountValueByProductIdResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**UpdateDiscountValueByProductId Sample Call**

**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 Sample Call (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());

**}**

### Update Discount Amount with Store Product Code (UpdateDiscountValueBySellerCode)

It is used to update discount information using the store product code of a product. The amount of discount entered is applied to the product listing price. As well as the difference between the list price and the discount price of the product, a discount is applied to the product stock unit prices.

**UpdateDiscountValueBySellerCodeRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Store product code* |
| ***discountType*** | [*See: DiscountType*](#_İndirim_Tipi_(Discount) |
| ***productDiscount.discountValue*** | *Discount amount (according to the parameter given in the discount type)* |
| *productDiscount.discountStartDate* | *Store discount start date* |
| *productDiscount.discountEndDate* | *Store discount end date* |

**\*Bold fields are mandatory fields.**

**UpdateDiscountValueBySellerCode Sample Call**

**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 Sample Call (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()); }

### Product Price Update with N11 Product Code (UpdateProductPriceById)

It is used to update only the product's base price information, product stock unit price information, or both, using the N11 product ID of a product.

**UpdateProductPriceByIdRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Product N11 ID* |
| *price* | *The price of the product to be updated* |
| *currencyType* | *Product listing price currency* |
| *product.stockItems.stockItem.sellerStockCode* | *Store product stock code* |
| *product.stockItems.stockItem.optionPrice* | *List price of the product stock unit* |

**\*Bold fields are mandatory fields.**

**UpdateProductPriceByIdResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**UpdateProductPriceById Sample Call**

**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>

***UpdateProductPriceById Sample Call (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());  
}

### Update Product Price with Store Product Code (UpdateProductPriceBySellerCode)

It is used to update the price information of a product using the store product code.

**UpdateProductPriceBySellerCodeRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Product code of the product on the store* |
| ***price*** | *Base price of the product to be updated* |
| *currencyType* | *Product listing price currency* |
| *product.stockItems.stockItem.sellerStockCode* | *Store product stock code* |
| *product.stockItems.stockItem.optionPrice* | *List price of the product stock unit* |

**\*Bold fields are mandatory fields.**

**UpdateProductPriceBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**UpdateProductPriceBySellerCode Sample Call**

**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 Sample Call (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();

}

### Updating Basic Product Attributes (UpdateProductBasic)

It is used to update a registered product's N11 product ID or store code and product price, discount on the product, and optionally the stock price specified by the stock ID or store stock code, and the stock price of the relevant stock unit.

**UpdateProductBasicRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| *productId* | *Product N11 ID information* |
| *productSellerCode* | *Product store code* |
| *price* | *Base price of the product* |
| *productDiscount.discountType* | [*See: DiscountType*](#_İndirim_Tipi_(Discount) |
| *productDiscount.discountValue* | *Discount Amount (according to the parameter given in the discount type)* |
| *productDiscount.discountStartDate* | *Store discount start date* |
| *productDiscount.discountEndDate* | *Store discount end date* |
| *stockItems.stockItem.sellerStockCode* | *Product stock store code* |
| *stockItems.stockItem.id* | *Product stock N11 ID* |
| *stockItems.stockItem.optionPrice* | *List price of the product stock unit* |
| *stockItems.stockItem.quantity* | *Product stock quantity* |
| *description* | *Product Description* |
| *images.image.order* | *Image priority number* |
| *images.image.url* | *Image url address* |

**\*Bold fields are mandatory fields.**

**UpdateProductBasicResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**UpdateProductBasic Sample Call**

**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>

<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>

<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 Sample Call (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());

}

### Listing Product Questions (GetProductQuestionList)

*Lists the questions asked by your customers to your store. Your Appkey and Appsecret information is required to list your questions.*

**GetProductQuestionListRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| *productQuestionSearch.productId* | *Product N11 ID* |
| *productQuestionSearch.buyerEmail* | *Customer e-mail address* |
| *productQuestionSearch.subject* | *Subject of the question* |
| *productQuestionSearch.status* | *Status of incoming questions* |
| *productQuestionSearch.questionDate* | *The date of the question* |
| ***currentPage*** | *Page number shown (currentpage starts with 0)* |
| ***pageSize*** | *Number of items on the page* |

**\*Bold fields are mandatory fields.**

**GetProductQuestionListResponse**

|  |  |
| --- | --- |
| *ProductQuestionId* | *Question ID on the N11 side of the incoming question.* |
| *productQuestion.productId* | *The ID that the product has on the N11* |
| *productQuestion.producTitle* | *Product title* |
| *productQuestion.question* | *Description of the incoming question* |
| *productQuestion.subject* | *Subject of the question* |
| *productQuestion.questionDate* | *The date of the question* |
| *productQuestion.answer* | *Response Content* |
| *currentPage* | *Page number shown (currentpage starts with 0)* |
| *pageSize* | *Number of items on the page* |
| *Result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |

**GetProductQuestionList Sample Call**

**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>

<answer>ccxccxxcxcxccxcxxcxc</answer>

<id>358708</id>

<productId>16527931</productId>

<productTitle>BundleProduct20 BundleProduct20</productTitle>

<question>sdagakdgjkdgkjdsagjkdsgakjgsdak</question>

<questionSubject>testfest</questionSubject>

</productQuestion>

</productQuestions>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

<totalCount>1</totalCount>

<pageCount>1</pageCount>

</pagingData>

</ns3:GetProductQuestionListResponse>

</env:Body>

</env:Envelope>

**GetProductQuestionList Sample Call (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());  
 }  
}

### Detailing Product Questions (GetProductQuestionDetail)

*You can view the contents of the questions listed with GetProductQuestionList with the help of the ID and getProductQuestionDetail.*

**GetProductQuestionDetailRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productQuestionId*** | *N11 id of Incoming Question* |

**\*Bold fields are mandatory fields.**

**GetProductQuestionDetailResponse**

|  |  |
| --- | --- |
| *Result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *productQuesiton.asnwer* | Response Content |
| *productQuestion.answeredDate* | *Response date* |
| *productQuestion.buyerExpose* | *Status of the incoming question* |
| *productQuestion.Email* | *Customer e-mail address* |
| *productQuestion.FullName* | *Customer name* |
| *productQuestion.productId* | *Product ID* |
| *productQuestion.productStatus* | *Product status* |
| *productQuestion.productTitle* | *Product title* |
| *productQuestion.question* | *Question description* |
| *productQuestion.questionDate* | *Question date* |
| *productQuestion.questionSubject* | *Subject of the question* |
| *productQuestion.sellerExpose* | *Status of the answered question (such as private, public)* |
| *Status* | *Status of the question* |

**GetProductQuestionDetail Sample Call**

**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>

<answer>Cevap İçeriği</answer>

<answeredDate>2014-01-09</answeredDate>

<buyerExpose>YES</buyerExpose>

<email>seleniummallfront@mailcatch.com</email>

<fullName>Selenium</fullName>

<productId>16527931</productId>

<productStatus>Active</productStatus>

<productTitle>BundleProduct20 BundleProduct20</productTitle>

<question>Soru İçeriği</question>

<questionDate>2014-01-09</questionDate>

<questionSubject>testfest</questionSubject>

<sellerExpose>YES</sellerExpose>

<status>CLOSED</status>

</productQuestion>

</ns3:GetProductQuestionDetailResponse>

</env:Body>

</env:Envelope>

**GetProductQuestionDetail Sample Call (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());  
}

### Answering Product Questions (SaveProductAnswer)

*It is used to answer product questions from customers. To answer, the productQuestionId value is mandatory and can be obtained from GetProductQuestionList.*

**SaveProductAnswerRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productQuestionId*** | *N11 ID of Incoming Question* |
| ***productAnswer*** | *Answer of the question* |

**\*Bold fields are mandatory fields.**

**SaveProductAnswerResponse**

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

**SaveProductAnswer Sample Call**

**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 Sample Call (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());  
}

### Get the Product-Status Number (ProductAllStatusCountsRequest)

*It sorts all products of the seller based on their statuses, and provides status / number information. Since the response is returned according to the seller, only authorization is sufficient in the request. Each seller can send a maximum of 3 requests.*

**ProductApprovalStatusRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |

**\*Bold fields are mandatory fields.**

**ProductApprovalStatusResponse**

|  |  |
| --- | --- |
| ***result*** | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| ***approvedCount*** | The number of products whose ProductApprovalUnionStatus values are active |
| ***unapprovedCount*** | Number of products with ProductApprovalstatus unapprovedUpdate |
| ***waitingApprovalCount*** | The number of products with ProductApprovalStatus value waitingForApproval or RejectedBefore |
| ***waitingCount*** | The number of products whose ProductApprovalUnionStatus values are Suspended |
| ***rejectedCount*** | The number of products whose ProductApprovalStatus values are Rejected |
| ***totalCount*** | Number of all products |

**ProductAllStatusCountsRequest Sample Call**

**Request**

<?xml version="1.0" encoding="UTF-8"?>

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

<soapenv:Header />

<soapenv:Body>

<sch:ProductApprovalStatusRequest>

<auth>

<appKey>c20f535e-5bf2-4d1e-825b-bfd321038fec</appKey>

<appSecret>Y1KLs2Utulukaxtu</appSecret>

</auth>

</sch:ProductApprovalStatusRequest>

</soapenv:Body>

</soapenv:Envelope>

***Response***

<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header />

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

<ProductApprovalStatusResponseResult>

<approvedCount>22</approvedCount>

<rejectedCount>3</rejectedCount>

<totalCount>55</totalCount>

<unapprovedCount>3</unapprovedCount>

<waitingApproval>27</waitingApproval>

<waitingCount>4</waitingCount>

</ProductApprovalStatusResponseResult>

</ns3:ProductApprovalStatusResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

***Response (When the limit is exceeded)***

<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header />

<SOAP-ENV:Body>

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

<result>

<status>failure</status>

<errorCode>SELLER\_API.getProductApprovalStatusCountRequest.

maxCallLimit.reached</errorCode>

<errorMessage>1 gün içerisinde gönderebileceğiniz istek sayısı en

fazla 3 tanedir.</errorMessage>

<errorCategory>SELLER\_API</errorCategory>

</result>

</ns3:ProductApprovalStatusResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ProductAllStatusCountsRequest Sample Call (JAVA)**

public static void main (String []args) {  
 String strAppKey = "\*\*\*";  
 String strAppSecret = "\*\*\*";  
 Authentication authentication = new Authentication();  
 authentication.setAppKey(strAppKey);  
 authentication.setAppSecret(strAppSecret);  
  
 ProductApprovalStatusRequest request = new ProductApprovalStatusRequest();  
 request.setAuth(authentication);  
  
 ProductServicePort port = new ProductServicePortService().getProductServicePortSoap11();  
 ProductApprovalStatusResponse response = port.productAllStatusCountsRequest(request);

ProductApprovalStatusApiModel model = response.getProductApprovalStatusApiModel();  
  
 System.*out*.println(model.getApprovedCount() + “,” + model.getUnapprovedCount() + “,” + model.getRejectedCount() + “,” + model.getWaitingApprovalCount() + “,” + model.getWaitingCount() + “,” + model.getTotalCount());  
}

## 3.4 Product Sales Services (ProductSellingService)

WSDL address for product sales services:

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

### Start Selling the Product with N11 Product ID (StartSellingProductByProductId)

It is used to start sales using N11 product ID of a product that is not on sale.

**StartSellingProductByProductIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Product ID on the n11 side of the product* |

**\*Bold fields are mandatory fields.**

**StartSellingProductByProductIdResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**StartSellingProductByProductId Sample Call**

**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 Sample Call (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.");  
}

### Start Selling the Product with the Store Product Code (StartSellingProductBySellerCode)

It is used to start selling a product that is not on sale using the store product code.

**StartSellingProductBySellerCodeRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Product code on the store side of the product* |

**\*Bold fields are mandatory fields.**

**StartSellingProductBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**StartSellingProductBySellerCode Sample Call**

**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 Sample Call (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.");

}

### Stop Selling the Product with N11 Product ID (StopSellingProductByProductId)

It is used to close the product on sale using the n11 product ID.

**StopSellingProductByProductIdRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productId*** | *Product code on the n11 side of the product* |

**\*Bold fields are mandatory fields.**

**StopSellingProductByProductIdResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**StopSellingProductByProductId Sample Call**

**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 Sample Call (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.");

}

### Stop Selling the Product with the Store Product Code (StopSellingProductBySellerCode)

It is used to stop the sale of the product on sale using the store product code.

**StopSellingProductBySellerCodeRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Product code on the store side of the product* |

**\*Bold fields are mandatory fields.**

**StopSellingProductBySellerCodeResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *products.product* | [*See: Product Summary Information*](#_Ürün_Özet_Bilgi) |

**StopSellingProductBySellerCode Sample Call**

**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 Sample Call (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 Product Stock Services (ProductStockService)

WSDL address for product stock services:

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

### Listing Stock Information with N11 Product ID (GetProductStockByProductId)

It is the method that retrieves N11 product ID and product stock information of the product registered in the system.

There is also “**version**” information of the stock status in the answer, if there is a change in product stocks, this version information will increase; the calling party can check the version information and see whether there is a change in the stock information given to N11.

***GetProductStockByProductIdRequest***

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***product.Id*** | *Product ID information* |

**\*Bold fields are mandatory fields.**

***GetProductStockByProductIdResponse***

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.id* | *Option id with stock definition* |
| *stockItems.stockItem.attributes.atribute.name* | *Option name with stock definition* |
| *stockItems.stockItem.attributes.atribute.value* | *Option value with stock definition* |
| *stockItems.stockItem.id* | *Product stock N11 ID* |
| *stockItems.stockItem.sellerCode* | *Product stock store code* |
| *stockItems.stockItem.quantity* | *Product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**GetProductStockByProductId Sample Call**

**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 Sample Call (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());

}

}

### Listing Stock Information with Store Product Code (GetProductStockBySellerCode)

It is the method that retrieves the store product code and product stock information of the product registered in the system.

There is also “**version**” information of the stock status in the answer, if there is a change in product stocks, this version information will increase; the calling party can check the version information and see whether there is a change in the stock information given to N11.

***GetProductStockBySellerCodeRequest***

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***productSellerCode*** | *Store product code* |

**\*Bold fields are mandatory fields.**

***GetProductStockBySellerCodeResponse***

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.id* | *Option id with stock definition* |
| *stockItems.stockItem.attributes.atribute.name* | *Option name with stock definition* |
| *stockItems.stockItem.attributes.atribute.value* | *Option value with stock definition* |
| *stockItems.stockItem.id* | *Product stock N11 ID* |
| *stockItems.stockItem.sellerCode* | *Product stock store code* |
| *stockItems.stockItem.quantity* | *Product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**GetProductStockBySellerCode Sample Call**

**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 Sample Call (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());

}

}

### Updating Stock Quantity with Stock Attributes (DeleteAndUpdateStockByStockAttributes )

It is used to update the stock information of the registered product using the product stock option information. Updating is done by entering the product n11 ID and stock option information. Stock information for a product can be accessed via the **GetProductStockByProductId** or **GetProductStockBySellerCode** methods in **ProductStockService**. It is possible to update all stock information for a product.

In order not to overwhelm the varying stock amounts on the N11 side, if "**version**" information is given, comparison is made with the version of the relevant product stock information on N11, and the operation is not performed if the stock version numbers are incompatible.

***DeleteAndUpdateStockByStockAttributesRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***product.Id*** | *Product ID information* |
| ***product.stockItems.stockItem.attributes.attribute.name*** | *Product stock attribute name* |
| ***product.stockItems.stockItem.attributes.attribute.value*** | *Product stock attribute value* |
| ***product.stockItems.stockItem.quantity*** | *Product stock quantity to be updated* |
| *product.stockItems.stockItem.version* | *Product version number* |

**\*Bold fields are mandatory fields.**

***DeleteAndUpdateStockByStockAttributesResponse***

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Option name with stock definition* |
| *stockItems.stockItem.attributes.atribute.value* | *Option value with stock definition* |
| *stockItems.stockItem.id* | *Product stock N11 ID* |
| *stockItems.stockItem.sellerStockCode* | *Product stock store code* |
| *stockItems.stockItem.stock* | *Product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**DeleteAndUpdateStockByStockAttributes Sample Call**

**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 Sample Call (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();

}

}

### Stock Update with N11 Product Stock ID (UpdateStockByStockId)

It is used to update the stock information of the registered product using the n11 product stock ID. n11 product inventory ID and quantity information is entered and updated. The n11 product stock ID of a product can be accessed using the **GetProductStockByProductId** or **GetProductStockBySellerCode** methods in **ProductStockService.**

In order not to overwhelm the varying stock amounts on the N11 side, if "**version**" information is given, comparison is made with the version of the relevant product stock information on N11, and the operation is not performed if the stock version numbers are incompatible.

With the UpdateStockByStockId and UpdateStockByStockSellerCode methods in the API, you can update up to 100 stock values in a single request for stock update operation.

Do not forget to make updates according to the product in order not to get an error.

For example, if the product you updated in the first 100 stocks is also included in the other 100 stocks, you will receive the "Product cannot be updated for 60 seconds" error because the system still continues the initial update. For this reason, you need to update the same product within the same 100 stock value.

***UpdateStockByStockId Request***

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***stockItems.stockItem.id*** | *Product stock N11 ID* |
| ***stockItems.stockItem.quantity*** | *New product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**\*Bold fields are mandatory fields.**

***UpdateStockByStockIdResponse***

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Option name with stock definition* |
| *stockItems.stockItem.attributes.atribute.value* | *Option value with stock definition* |
| *stockItems.stockItem.id* | *Product stock N11 ID* |
| *stockItems.stockItem.sellerStockCode* | *Product stock store code* |
| *stockItems.stockItem.stock* | *Product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**UpdateStockByStockId Sample Call**

**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 Sample Call (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());

}

### Stock Update with Store Stock Code (UpdateStockByStockSellerCode)

It is used to update the stock information of the registered product using the store product stock code. Updating is done by entering the store product inventory code and quantity information.

In order not to overwhelm the varying stock amounts on the N11 side, if "**version**" information is given, comparison is made with the version of the relevant product stock information on N11, and the operation is not performed if the stock version numbers are incompatible.

With the UpdateStockByStockId and UpdateStockByStockSellerCode methods in the API, you can update up to 100 stock values in a single request for stock update operation.

Do not forget to make updates according to the product in order not to get an error.

For example, if the product you updated in the first 100 stocks is also included in the other 100 stocks, you will receive the "Product cannot be updated for 60 seconds" error because the system still continues the initial update. For this reason, you need to update the same product within the same 100 stock value.

***UpdateStockByStockSellerCodeRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***stockItems.stockItem.sellerStockCode*** | *Product stock store code* |
| ***stockItems.stockItem.quantity*** | *New product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**\*Bold fields are mandatory fields.**

***UpdateStockByStockSellerCodeResponse***

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Option name with stock definition* |
| *stockItems.stockItem.attributes.atribute.value* | *Option value with stock definition* |
| *stockItems.stockItem.id* | *Product stock N11 ID* |
| *stockItems.stockItem.sellerStockCode* | *Product stock store code* |
| *stockItems.stockItem.stock* | *Product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**UpdateStockByStockSellerStockCode Sample Call**

**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 Sample Call (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());

### Increasing Stock Quantity with Stock Attributes (IncreaseStockByStockAttributes)

It is used to increase the stock amount of a product by using the stock option information.

In order not to overwhelm the varying stock amounts on the N11 side, if "**version**" information is given, comparison is made with the version of the relevant product stock information on N11, and the operation is not performed if the stock version numbers are incompatible.

***IncreaseStockByStockAttributesRequest***

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***product.id*** | *Product N11 ID* |
| ***product.stockItems.stockItem.attributes.attribute.name*** | *Option name with stock definition* |
| ***product.stockItems.stockItem.attributes.attribute.value*** | *Option value with stock definition* |
| ***product.stockItems.stockItem.quantityToIncrease*** | *Amount of stock to be increased* |
| *product.stockItems.stockItem.version* | *Product stock version number* |

**\*Bold fields are mandatory fields.**

***IncreaseStockByStockAttributesResponse***

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Option name with stock definition* |
| *stockItems.stockItem.attributes.atribute.value* | *Option value with stock definition* |
| *stockItems.stockItem.id* | *Product stock N11 ID* |
| *stockItems.stockItem.sellerStockCode* | *Product stock store code* |
| *stockItems.stockItem.stock* | *Product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**IncreaseStockByStockAttributes Sample Call**

**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 Sample Call (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());

}

### 

### Increasing Stock Quantity with N11 Product Stock ID (IncreaseStockByStockId)

It is used to increase the stock amount of a product using n11 product stock ID information. The n11 product stock ID of a product can be accessed using the **GetProductStockByProductId** or **GetProductStockBySellerCode** methods in **ProductStockService.**

In order not to overwhelm the varying stock amounts on the N11 side, if "**version**" information is given, comparison is made with the version of the relevant product stock information on N11, and the operation is not performed if the stock version numbers are incompatible.

***IncreaseStockByStockIdRequest***

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***stockItems.stockItem.id*** | *Product stock id information* |
| ***stockItems.stockItem.quantityToIncrease*** | *Amount of stock to be increased* |
| *product.stockItems.stockItem.version* | *Product version number* |

**\*Bold fields are mandatory fields.**

***IncreaseStockByStockIdResponse***

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Option name with stock definition* |
| *stockItems.stockItem.attributes.atribute.value* | *Option value with stock definition* |
| *stockItems.stockItem.id* | *Product stock item N11 ID* |
| *stockItems.stockItem.sellerCode* | *Product stock store code* |
| *stockItems.stockItem.quantity* | *Product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**IncreaseStockByStockId Sample Call**

**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 Sample Call (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());  
  
}

### Increasing Stock Quantity with Store Stock Code (IncreaseStockByStockSellerCode)

It is used to increase the stock amount of a product by using the store stock code.

In order not to overwhelm the varying stock amounts on the N11 side, if "**version**" information is given, comparison is made with the version of the relevant product stock information on N11, and the operation is not performed if the stock version numbers are incompatible.

***IncreaseStockByStockSellerCodeRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***stockItems.stockItem.sellerStockCode*** | *Store stock code* |
| ***stockItems.stockItem.quantityToIncrease*** | *Amount of stock to be increased* |
| *product.stockItems.stockItem.version* | *Product stock version number* |

**\*Bold fields are mandatory fields.**

***IncreaseStockByStockSellerCodeResponse***

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *stockItems.stockItem.attributes.atribute.name* | *Option name with stock definition* |
| *stockItems.stockItem.attributes.atribute.value* | *Option value with stock definition* |
| *stockItems.stockItem.id* | *Product stock N11 ID* |
| *stockItems.stockItem.sellerStockCode* | *Product stock store code* |
| *stockItems.stockItem.stock* | *Product stock quantity* |
| *stockItems.stockItem.version* | *Product stock version number* |

**IncreaseStockByStockSellerCode Sample Call**

**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 Sample Call (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 Order Services (OrderService)

WSDL address for order services: <https://api.n11.com/ws/OrderService.wsdl>

### Detailed Order Listing (DetailedOrderList)

It is used to list the order items together with the order information according to the given search criteria.

**DetailedOrderListRequest**

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| *searchData.productId* | *Product N11 ID* |
| *searchData.status* | [*See: SearchData*](#_Sipariş_Durumu_(SearchData)) |
| *searchData.buyerName* | *Ordering N11 member’s name* |
| *searchData.orderNumber* | *Order number* |
| *searchData.productSellerCode* | *Store code of the product* |
| *searchData.recipient* | *The recipient's name* |
| *searchData.period.startDate* | *Order creation start date* |
| *searchData.period.endDate* | *Order creation end date* |
| *searchData.sortForUpdateDate* | *Lists orders by UpdateDate* [See: Listing](#SortForUpdate) |
| *pagingData* | [*See: Pagination*](#_Sayfalandırma_(Pagination)_1) |

**\*Bold fields are mandatory fields.**

**DetailedOrderListResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *pagingData* | [*See: Pagination*](#_Sayfalandırma_(Pagination)_1) |
| *orderList.order.citizenshipId* | *Recipient TC identification number* |
| *orderList.order.createDate* | *Order creation date* |
| *orderList.order.id* | *N11 ID of the Order* |
| *orderList.order.orderNumber* | *Order number* |
| *paymentType* | [*See: Payment Type*](#_Ödeme_Tipi_(Payment) |
| *orderList.order.totalAmount* | *Total amount of the order* |
| *orderList.order.totalDiscountAmount* | *Total discount amount of the order* |
| *orderList.order.orderItemList.orderItem* | *Order items list* |
| *orderList.order.orderItemList.orderItem.commission* | *N11 service amount of the order item* |
| *deliveryFeeType* | [*See: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderList.order.orderItemList.orderItem.dueAmount* | *Amount to be collected* |
| *orderList.order.orderItemList.orderItem.id* | *Order item ID* |
| *orderList.order.orderItemList.orderItem.installmentChargeWithVAT* | *Maturity difference (including VAT)* |
| *orderList.order.orderItemList.orderItem.mallDiscount* | *N11 discount on the product* |
| *orderList.order.orderItemList.orderItem.price* | *Unit price of the product (excluding discounts)* |
| *orderList.order.orderItemList.orderItem.productId* | *Product N11 ID* |
| *orderList.order.orderItemList.orderItem.productName* | *Product name* |
| *orderList.order.orderItemList.orderItem.productSellerCode* | *Product store code* |
| *orderList.order.orderItemList.orderItem.quantity* | *Order quantity* |
| *orderList.order.orderItemList.orderItem.sellerCouponDiscount* | *Store coupon discount* |
| *orderList.order.orderItemList.orderItem.sellerDiscount* | *Store discount* |
| *orderList.order.orderItemList.orderItem.sellerInvoiceAmount* | *Store invoice amount* |
| *orderList.order.orderItemList.orderItem.shipmentInfo.shipmentCompany.id* | *Shipping company id information* |
| *orderList.order.orderItemList.orderItem.shipmentInfo.shipmentCompany.name* | *Shipping company name* |
| *orderList.order.orderItemList.orderItem.shipmentInfo.shipmentMethod* | *If the shipping method is 1, the cargo if 2 the other* |
| *orderList.order.orderItemList.orderItem.shipmentDate* | *Shipment date* |
| *orderItem.status* | [*See: OrderItem Status*](#_Sipariş_Maddesi_Durum) |
| *orderList.order.orderItemList.orderItem.totalMallDiscountPrice* | *Sum of N11 discounts* |
| *orderList.order.orderItemList.orderItem.version* | *Stock tracking version number of the order item* |
| *orderList.order.orderNumber* | *Order number* |
| *orderList.order.paymentType* | [*See: Payment type*](#_Ödeme_Tipi__1) |
| *orderList.order.status* | [*See: Order Status*](#_Sipariş_Durum_Bilgisi) |
| *orderList.order.totalAmount* | *Order amount* |
| *orderList.order.totalDiscountAmount* | *Order total discount amount (Seller+n11)* |

**DetailedOrderList Sample Call**

**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></productId>

<status></status>

<buyerName></buyerName>

<orderNumber></orderNumber>

<productSellerCode></productSellerCode>

<recipient></recipient>

<sameDayDelivery></sameDayDelivery>

<period>

<startDate>20/02/2018</startDate>

<endDate>21/02/2018</endDate>

</period>

<sortForUpdateDate>true</sortForUpdateDate>

</searchData>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

<totalCount></totalCount>

<pageCount></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 Sample Call (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());

}

}

### Order Listing (OrderList)

*This method is used to list summary information about the order.*

***OrderListRequest***

|  |  |
| --- | --- |
| ***auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| *searchData.productId* | *Product N11 ID* |
| *searchData.status* | [*See: SearchData*](#_Sipariş_Durumu_(SearchData)) |
| *searchData.buyerName* | *Ordering N11 member’s name* |
| *searchData.orderNumber* | *Order number* |
| *searchData.productSellerCode* | *Store code of the product* |
| *searchData.recipient* | *The recipient's name* |
| *searchData.period.startDate* | *Order creation start date* |
| *searchData.period.endDate* | *Order creation end date* |
| *searchData.sortForUpdateDate* | *Lists Updated Orders* [See: Listing](#SortForUpdate) |
| *pagingData* | [*See: Pagination*](#_Sayfalandırma_(Pagination)_1) |

**\*Bold fields are mandatory fields.**

**OrderListResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderList.order.citizenshipId* | *TC identification number* |
| *orderList.order.createDate* | *Order creation date* |
| *orderList.order.id* | *Order n11 ID information* |
| *orderList.order.orderNumber* | *Order number* |
| *paymentType* | [*See: Payment Type*](#_Ödeme_Tipi_(Payment) |
| *orderList.order.status* | [*See: Order Status*](#_Sipariş_Durum_Bilgisi_1) |

**OrderList Sample Call**

**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></status>

<buyerName></buyerName>

<orderNumber></orderNumber>

<productSellerCode></productSellerCode>

<recipient></recipient>

<sameDayDelivery></sameDayDelivery>

<period>

<startDate></startDate>

<endDate></endDate>

</period>

<sortForUpdateDate>false</sortForUpdateDate>

</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 Sample Call (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());

}

}

### Order Detail (OrderDetail)

*Order N11 ID information is used to get order details, order N11 ID information can be accessed via OrderService OrderList or DetailedOrderList methods.*

*Payment of the shipping fee and its collection information can be accessed from the “serviceItemList” field via the n11 platform.*

***OrderDetailRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***orderRequest.id*** | *N11 ID of the Order* |

**\*Bold fields are mandatory fields.**

**OrderDetailResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderDetail.citizenshipId* | *TC identification number* |
| *orderDetail.createDate* | *Order creation date* |
| *orderDetail.id* | *N11 ID of the Order* |
| *orderDetail.orderNumber* | *Order number* |
| *paymentType* | [*See: Payment Type*](#_Ödeme_Tipi_) |
| *orderDetail.billingAddress.address* | *Billing address information* |
| *orderDetail.billingAddress.city* | *Address city* |
| *orderDetail.billingAddress.district* | *Address district* |
| *orderDetail.billingAddress.fullName* | *Address name* |
| *orderDetail.billingAddress.gsm* | *Recipient phone number* |
| *orderDetail.billingAddress.neighborhood* | *Address district/neighborhood* |
| *orderDetail.billingAddress.postalCode* | *Address zip code* |
| *orderDetail.billingAddress.taxId* | *Tax number* |
| *orderDetail.billingAddress. taxOffice* | *Tax administration* |
| *orderDetail.billingAddress.tcId* | *TC Id* |
| *orderDetail.billingTemplate.dueAmount* | *Maturity difference* |
| *ordelDetail.billingTemplate.installmentChargeWithVat* | *Maturity difference including VAT* |
| *orderDetail.billingTemplate.originalPrice* | *Price of the Order Item before all discounts* |
| *orderDetail.billingTemplate.sellerInvoiceAmount* | *Commission invoice amount* |
| *orderDetail.billingTemplate.totalMallDiscountPrice* | *All N11 discounts* |
| *orderDetail.billingTemplate.totalSellerDiscount* | *Total Seller discount* |
| *orderDetail.billingTemplate.totalServiceItemOriginalPrice* | *Total Service Price* |
| *orderDetail.buyer.fullName* | *Purchaser name of the order* |
| *orderDetail.buyer.id* | *Purchaser order N11 ID* |
| *orderDetail.buyer.taxId* | *Tax number* |
| *orderDetail.buyer.taxOffice* | *Tax administration* |
| *orderDetail.buyer.tcId* | *TC Id* |
| *orderDetail.buyer.email* | *E-mail of the purchaser of the order* |
| *orderDetail.invoiceType* | *Bill type: 1: Individual*  *2: Corporate* |
| *orderDetail.itemList.item.approveDate* | *Order item accepted date* |
| *orderDetail.itemList.item.attributes.attribute.name* | *Order item attribute name* |
| *orderDetail.itemList.item.attributes.attribute.value* | *Order item attribute value* |
| *orderDetail.itemList.item.commission* | *N11 service amount of the order item* |
| *orderDetail.itemList.item.customTextOptionsValues* | *Additional information entered by the customer for the order item* |
| *orderDetail.itemList.item.sellerInvoiceAmount* | *Vendor invoice amount (the amount of invoice that the seller has to bill for an order)* |
| *orderDetail.itemList.item.deliveryFeeType* | [*See: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderDetail.itemList.item.dueAmount* | *Amount to be collected* |
| *orderDetail.itemList.item.id* | *Order item’s n11 ID* |
| *orderDetail.itemList.item.installmentChargeWithVAT* | *Maturity difference (including VAT)* |
| *orderDetail.itemList.item.mallDiscount* | *N11 discount on the product* |
| *orderDetail.itemList.item.price* | *Unit price of the product (list price, excluding discounts)* |
| *orderDetail.itemList.item.productId* | *Product N11 ID* |
| *orderDetail.itemList.item.productName* | *Product name* |
| *orderDetail.itemList.item.productSellerCode* | *Product store code* |
| *orderDetail.itemList.item.quantity* | *Order quantity* |
| *orderDetail.itemList.item.sellerCouponDiscount* | *Store coupon discount* |
| *orderDetail.itemList.item.sellerDiscount* | *Store discount* |
| *orderDetail.itemList.item.shipmentInfo.campaignNumberStatus* | *Special offer number status;*  *1: USED (Used) 2: UNUSED (Unused)* |
| *orderDetail.itemList.item.shipmentInfo.shipmentCode* | *Shipment code* |
| *orderDetail.itemList.item.shipmentInfo.shipmentCompany.id* | *Shipping company id information* |
| *orderDetail.itemList.item.shipmentInfo.shipmentCompany.name* | *Shipping company name* |
| *orderDetail.itemList.item.shipmentInfo.shipmentCompany.shortName* | *Shipping company’s short name* |
| *orderDetail.itemList.item.shipmentInfo.shipmentMethod* | *Shipment method 1: Shipment 2: Other* |
| *orderDetail.itemList.item.shipmentDate* | *Cargo date* |
| *orderDetail.itemList.item.status* | [*See: OrderItem Status*](#_Sipariş_Maddesi_Durum) |
| *orderDetail.itemList.item.totalMallDiscountPrice* | *Sum of n11 discounts* |
| *orderDetail.itemList.item.version* | *Stock version number of the relevant product* |
| *orderDetail.serviceItemList.serviceItem.intallmentChargeWithVAT* | *Maturity difference including service vat* |
| *orderDetail.serviceItemList.serviceItem.orderItemList.orderItem* | *Service order item id* |
| *orderDetail.serviceItemList.serviceItem.orderItemType* | [*See: Service order item type*](#_Sipariş_Kalemi_tipi) |
| *orderDetail.serviceItemList.serviceItem.price* | *Service item price* |
| *orderDetail.serviceItemList.serviceItem.quantity* | *Number of service* |
| *orderDetail.serviceItemList.serviceItem.sellerInvoiceAmount* | *Service invoice amount* |
| *orderDetail.serviceItemList.serviceItem.totalDiscountPrice* | *Total discount amount of the service* |
| *orderDetail.shippingAddress.address* | *Shipment address information* |
| *orderDetail.shippingAddress.city* | *Shipment address city* |
| *orderDetail.shippingAddress.district* | *Shipment address district* |
| *orderDetail.shippingAddress.fullName* | *Shipment address name* |
| *orderDetail.shippingAddress.gsm* | *Shipment address phone no* |
| *orderDetail.shippingAddress.neighborhood* | *Shipment district/neighborhood* |
| *orderDetail.shippingAddress.postalCode* | *Shipment address zip code* |

**OrderDetail Sample Call**

**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>

<neighborhood>Sarıyer Mh.</neighborhood>

<postalCode>\*\*\*\*</postalCode> <taxId/>

<taxOffice/>

<tcId/>

</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 Sample Call (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());

}

}

### Acceptance of Order Item (OrderItemAccept)

It is used to accept the order item using n11 ID. The accepted order cannot be rejected by the store later. The order n11 ID is accessed from the **OrderService** using the **OrderDetail** or **DetailedOrderList** method.

***OrderItemAcceptRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***orderItem.id*** | *Order item n11 ID* |

**\*Bold fields are mandatory fields.**

***OrderItemAcceptResponse***

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderItemList.orderItem.aprovedDate* | *Order confirmation date* |
| *orderItemList.orderItem.attributes.attribute.name* | *Order item attribute name* |
| *orderItemList.orderItem.attributes.attribute.value* | *Order item attribute value* |
| *orderItemList.orderItem.commission* | *N11 service amount related to the order item* |
| *orderItemList.orderItem.customTextOptionValues* | *Attribute value entered in the order item* |
| *orderItemList.orderItem.deliveryFeeType* | [*See: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderItemList.orderItem.dueAmount* | *Amount to be collected* |
| *orderItemList.orderItem.id* | *Order item n11 ID information* |
| *orderItemList.orderItem.installmentChargeWithVAT* | *Maturity difference (including VAT)* |
| *orderItemList.orderItem.mallDiscount* | *N11 discount on the product* |
| *orderItemList.orderItem.price* | *Unit price of the product (excluding discounts)* |
| *orderItemList.orderItem.productId* | *Product N11 ID* |
| *orderItemList.orderItem.productName* | *Product name* |
| *orderItemList.orderItem.productSellerCode* | *Product store code* |
| *orderItemList.orderItem.quantity* | *Order quantity* |
| *orderItemList.orderItem.sellerCouponDiscount* | *Store coupon discount* |
| *orderItemList.orderItem.sellerDiscount* | *Store discount* |
| *orderItemList.orderItem.sellerInvoiceAmount* | *Store invoice amount* |
| *orderItemList.orderitem.shipmentInfo.campaignNumber* | *Special Offer Number* |
| *orderItemList.orderitem.shipmentInfo.campaignNumberStatus* | *Special Offer Number Status*  *1: Used 2:Unused* |
| *orderItemList.orderitem.shipmentInfo.shipmentCode* | *Shipment code* |
| *orderItemList.orderitem.shipmentInfo.shipmentCompany.id* | *Shipping company id information* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.name* | *Shipping company name* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.shortName* | *Shipping company’s short name* |
| *orderItem.shipmentInfo.shipmentMethod* | *Shipment method*  *1: Cargo*  *2: Other* |
| *orderItem.shipmentDate* | *Cargo date* |
| *orderItem.status* | [*See: OrderItem Status*](#_Sipariş_Maddesi_Durum) |
| *orderItem.totalMallDiscountPrice* | *Sum of N11 discounts* |
| *orderItem.version* | *Stock version number of the relevant product* |

**OrderItemAccept Sample Call**

**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 Sample Call (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();

}

### Rejection of Order Item (OrderItemReject)

It is used to reject the order item using the n11 ID. The rejected order cannot be accepted by the store later. The order n11 ID is accessed from the **OrderService** using the **OrderDetail** or **DetailedOrderList** method.

***OrderItemRejectRequest***

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***orderItemList.orderItem.id*** | *Order item n11 ID information* |
| ***rejectReason*** | *Reason for order rejection* |
| ***rejectReasonType*** | *Order rejection type:*  *OUT\_OF\_STOCK (Out of stock), OTHER (Other)* |

**\*Bold fields are mandatory fields.**

**OrderItemRejectResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderItemList.orderItem.approvedDate* | *Order confirmation date* |
| *orderItemList.orderItem.attributes* | *Attributes of the order item* |
| *orderItemList.orderItem.commission* | *N11 service amount related to the order item* |
| *orderItemList.orderItem.customTextOptionsValues* | *Additional information entered by the customer for the order item* |
| *orderItemList.orderItem.deliveryFeeType* | [*See: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderItemList.orderItem.dueAmount* | *Amount to be collected* |
| *orderItemList.orderItem.id* | *Order item n11 ID information* |
| *orderItemList.orderItem.installmentChargeWithVAT* | *Maturity difference (including VAT)* |
| *orderItemList.orderItem.mallDiscount* | *N11 discount on the product* |
| *orderItemList.orderItem.price* | *Unit price of the product (excluding discounts)* |
| *orderItemList.orderItem.productId* | *Product N11 ID* |
| *orderItemList.orderItem.productName* | *Product name* |
| *orderItemList.orderItem.productSellerCode* | *Product store code* |
| *orderItemList.orderItem.quantity* | *Order quantity* |
| *orderItemList.orderitem.rejectReason* | *Reason for order rejection* |
| *orderItemList.orderitem.rejectReasonType* | *Order rejection type* |
| *orderItemList.orderItem.sellerCouponDiscount* | *Store coupon discount* |
| *orderItemList.orderItem.sellerDiscount* | *Store discount* |
| *orderItemList.orderItem.sellerInvoiceAmount* | *Store invoice amount* |
| *orderItemList.orderItem.shipmentInfo.campaignNumberStatus* | *Special offer number status;*  *1: USED (Used) 2: UnUSED (Unused)* |
| *orderItemList.orderItemshipmentInfo.shipmentCode* | *Shipment code* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.id* | *Shipping company id information* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.name* | *Shipping company name* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.shortName* | *Shipping company’s short name* |
| *orderItemList.orderItem.shipmentInfo.shipmentMethod* | *Shipment method*  *1: Cargo*  *2: Other* |
| *orderItemList.orderItem.shippingDate* | *Cargo date* |
| *orderItemList.orderItem.status* | [*See: Orderitem Status*](#_Sipariş_Maddesi_Durum) |
| *orderItemList.orderItem.totalMallDiscountPrice* | *Sum of N11 discounts* |
| *orderItemList.orderItem.version* | *Stock version number of the relevant product* |

**OrderItemReject Sample Call**

**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 Sample Call (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();

}

### Shipment of the Order Item (MakeOrderItemShipment)

It is used to ship the order item using n11 ID. The order n11 ID is accessed from the **OrderService** using the **OrderDetail** or **DetailedOrderList** method.

The status of the order item (OrderItemStatus) that needs to be updated as a prerequisite should be "**Paid**" or "**Accepted**". Otherwise, the answer "**preconditions could not be met**" is received.

**GetShipmentCompanies** method from **ShipmentCompanyService** should be used for the list of shipping companies.

**MakeOrderItemShipmentRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authorization*](#_Yetkilendirme_(Authentication)) |
| ***orderItemList.orderItem.id*** | *Product N11 ID* |
| ***orderItemList.orderItem.shipmentInfo.shipmentCompany.id*** | *Shipping company's n11 ID* |
| *orderItemList.orderItem.shipmentInfo.campaignNumber* | *Order special offer number* |
| ***orderItemList.orderItem.shipmentInfo.trackingNumber*** | *Tracking number* |
| ***orderItemList.orderItem.shipmentInfo.shipmentMethod*** | *Shipment method*  *1: Cargo*  *2: Other* |

**\*Bold fields are mandatory fields.**

**MakeOrderItemShipmentResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *orderItemList.orderItem.approvedDate* | *Order confirmation date* |
| *orderItemList.orderItem.attributes* | *Order item attributes* |
| *orderItemList.orderItem.commission* | *N11 service amount related to the order item* |
| *orderItemList.orderItem.customTextOptionValues* | *Additional information entered by the customer for the order item* |
| *orderItemList.orderItem.deliveryFeeType* | [*See: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *orderItemList.orderItem.dueAmount* | *Amount to be collected* |
| *orderItemList.orderItem.id* | *Order item n11 ID information* |
| *orderItemList.orderItem.installmentChargeWithVAT* | *Maturity difference (including VAT)* |
| *orderItemList.orderItem.mallDiscount* | *N11 discount on the product* |
| *orderItemList.orderItem.price* | *Unit price of the product (excluding discounts)* |
| *orderItemList.orderItem.productId* | *Product N11 ID* |
| *orderItemList.orderItem.productName* | *Product name* |
| *orderItemList.orderItem.productSellerCode* | *Product store code* |
| *orderItemList.orderItem.quantity* | *Order quantity* |
| *orderItemList.orderItem.sellerCouponDiscount* | *Store coupon discount* |
| *orderItemList.orderItem.sellerDiscount* | *Store discount* |
| *orderItemList.orderItem.sellerInvoiceAmount* | *Store invoice amount* |
| *orderItemList.orderItem.shipmentInfo.campaignNumberStatus* | *Special offer number status;*  *1: USED (Used) 2: UNUSED (Unused)* |
| *orderItemList.orderItem.shipmentInfo.shipmentCode* | *Shipment code* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.id* | *Shipping company id information* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.name* | *Shipping company name* |
| *orderItemList.orderItem.shipmentInfo.shipmentCompany.shortName* | *Shipping Company’s short name* |
| *orderItemList.orderItem.shipmentInfo.shipmentMethod* | *Shipment method*  *1: Cargo*  *2: Other* |
| *orderItemList.orderItem.shipmentInfo.trackingNumber* | *Tracking number* |
| *orderItemList.orderItem.shippingDate* | *Cargo date* |
| *orderItemList.orderItem.status* | [*See: OrderItem Status*](#_Sipariş_Maddesi_Durum) |
| *orderItemList.orderItem.totalMallDiscountPrice* | *Sum of N11 discounts* |
| *orderItemList.orderItem.version* | *Stock version number of the relevant product* |

**MakeOrderItemShipment Sample Call**

**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 Sample Call (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 Shipping Company Service (ShipmentCompanyService)

WSDL address for shipping company service:

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

### Listing the Shipping Companies (GetShipmentCompanies)

Lists all the shipping companies defined in the system.

Since it is a public service, no security check is performed during the use of the service.

**GetShipmentCompaniesResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *shipmentCompanies.shipmentCompany.id* | *Shipping company id information* |
| *shipmentCompanies.shipmentCompany.name* | *Shipping company name* |
| *shipmentCompanies.shipmentCompany.shortName* | *Shipping company’s short name* |

**GetShipmentCompanies Sample Call**

**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 Sample Call (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 Shipment Template Service (ShipmentService)](#_3.7_Kargolama_Servisi)

WSDL address for shipment template service: <https://api.n11.com/ws/ShipmentService.wsdl>

### [Shipment Template Listing (GetShipmentTemplate)](#_Kargo_Şirketlerini_Listeleme)

Showing the attributes of the template searched with the shipment template name, such as the address method. deliverableCities means that the area we select the cities to be delivered to will be sent to all cities if no value is entered in this field.

**GetShipmentTemplateRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***name*** | *Shipment template name* |

**GetShipmentTemplateResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *shipmentTemplate.claimShipmentCompany.name* | *Return/Exchange shipping company’s name* |
| *shipmentTemplate.claimShipmentCompany.shortName* | *Return/Exchange shipping company’s short name* |
| *shipmentTemplate.combinedShipmentAllowed* | [*See: Shipment Allowed*](#_Kargo_Çoklu_Gönderim) |
| *shipmentTemplate.deliverableCities.city.code* | *Code of the city where delivery is made* |
| *shipmentTemplate.deliverableCities.city.name* | *Name of the city where delivery is made* |
| *shipmentTemplate.deliveryFeeType* | *If the shipping payment method is 1, buyer pays; if it’s 2 the store pays; if it’s 3 conditional shipment; if it’s 4, N11 pays* |
| *shipmentTemplate.exchangeAddress.address* | *Return address description* |
| *shipmentTemplate.exchangeAddress.city.code* | *Return address city code* |
| *shipmentTemplate.exchangeAddress.city.name* | *Return address city name* |
| *shipmentTemplate.exchangeAddress.district.id* | *Return address district id* |
| *shipmentTemplate.exchangeAddress.district.name* | *Return address district name* |
| *shipmentTemplate.exchangeAddress.postalCode* | *Return address zip code* |
| *shipmentTemplate.exchangeAddress.title* | *Return address title name* |
| *shipmentTemplate.exchangeAddress.feeCondition* | *Condition type of conditional shipment:*  *1:ByPrice 2: ByUnit* |
| *shipmentTemplate.exchangeAddress.feeConditionPrice/feeConditionUnit* | *Condition price/quantity of conditional shipment* |
| *shipmentTemplate.exchangeInfo* | *Exchange content* |
| *shipmentTemplate.shipmentCompanies.shipmentCompany.name* | *Shipping company name* |
| *shipmentTemplate.shipmentCompanies.shipmentCompany.shortName* | *Shipping company’s short name* |
| *shipmentTemplate.shipmentMethod* | *If the way of reaching the product is 1, the shipment; if 2, the other* |
| *shipmentTemplate.shippingInfo* | *Shipment information/text information will be entered* |
| *shipmentTemplate.specialDelivery* | *Special shipment true/false* |
| *shipmentTemplate.templateName* | *Shipment template* |
| *shipmentTemplate.warehouseAddess.address* | *Warehouse address information* |
| *shipmentTemplate.warehouseAddess.district.name* | *Warehouse district name* |
| *shipmentTemplate.warehouseAddess.district.id* | *Warehouse address district id* |
| *shipmentTemplate.warehouseAddess.feeCondition* | *Condition type of conditional shipment:*  *1:ByPrice 2: ByUnit* |
| *shipmentTemplate.warehouseAddess.feeConditionPrice* | *Condition price/quantity of conditional shipment* |
| *shipmentTemplate.warehouseAddess.city.name* | *Warehouse address city name* |
| *shipmentTemplate.warehouseAddess.city.code* | *Warehouse address city code* |
| *shipmentTemplate.warehouseAddess.city.postalCode* | *Warehouse address zip code* |
| *shipmentTemplate.warehouseAddess.title* | *Warehouse address title* |

**GetShipmentTemplate Sample Call**

**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 Sample Call (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());}}

### [Creating and Updating Shipment Templates (CreateOrUpdateShipmentTemplate)](#_Kargo_Şirketlerini_Listeleme)

The shipment template is a template created for how the shipment will go. The shipment template to be used in the orders is created through this service together with its attributes. claimshipmentCompany Use the <useDmallCargo> field, i.e. "1" is required.

**CreateOrUpdateShipmentTemplateRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***shipment.templateName*** | *Shipment template* |
| ***shipment.installmentInfo*** | *Setup information/text information will be entered* |
| ***shipment.exchangeInfo*** | *Change information/text information will be entered* |
| ***shipment.shippingInfo*** | *Shipment information/text information will be entered* |
| ***shipment.specialDelivery*** | *Special shipment true/false* |
| ***shipment.deliveryFeeType*** | *If the shipping payment method is 1, buyer pays; if it’s 2 the store pays; if it’s 3 conditional shipment* |
| *shipment.combinedShipmentAllowed* | [*See: Shipment Allowed*](#_Kargo_Çoklu_Gönderim) |
| ***shipment.shipmentMethod*** | *If the way of reaching the product is 1, the shipment; if 2, the other* |
| ***shipment.warehouseAddess.title*** | *Warehouse address title* |
| ***shipment.warehouseAddess.address*** | *Warehouse address information* |
| ***shipment.warehouseAddess.district.name*** | *Warehouse district name* |
| ***shipment.warehouseAddess.district.id*** | *Warehouse address district id* |
| ***shipment.warehouseAddess.city.name*** | *Warehouse address city name* |
| ***shipment.warehouseAddess.city.code*** | *Warehouse address city code* |
| ***shipment.warehouseAddess.city.postalCode*** | *Warehouse address zip code* |
| ***shipment.exchangeAddress.address*** | *Return address description* |
| ***shipment.exchangeAddress.city.code*** | *Return address city code* |
| ***shipment.exchangeAddress.city.name*** | *Return address city name* |
| ***shipment.exchangeAddress.district.id*** | *Return address district id* |
| ***shipment.exchangeAddress.district.name*** | *Return address district name* |
| ***shipment.exchangeAddress.postalCode*** | *Return address zip code* |
| ***shipment.exchangeAddress.title*** | *Return address title name* |
| ***shipment.shipmentCompanies.shipmentCompany.name*** | *Shipping company name* |
| ***shipment.shipmentCompanies.shipmentCompany.shortName*** | *Shipping company’s short name* |
| *shipmentTemplate.deliverableCities.city.code* | *Code of the city where delivery is made* |
| *shipmentTemplate.deliverableCities.city.name* | *Name of the city where delivery is made* |
| ***shipmentTemplate.claimShipmentCompany.name*** | *Return/Exchange shipping company’s name* |
| ***shipmentTemplate.claimShipmentCompany.shortName*** | *Return/Exchange shipping company’s short name* |
| *shipment.cargoAccountNo* | *Shipping account number* |
| ***shipment.useDmallCargo*** | *Use of n11.com integrated shipping company:*  *1: use*  *0: do not use* |

**\*Bold fields are mandatory fields.**

**CreateOrUpdateShipmentTemplateResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *shipmentTemplate.claimShipmentCompany.name* | *Return/Exchange shipping company’s name* |
| *shipmentTemplate.claimShipmentCompany.shortName* | *Return/Exchange shipping company’s short name* |
| *shipmentTemplate.combinedShipmentAllowed* | [*See: Shipment Allowed*](#_Kargo_Çoklu_Gönderim) |
| *shipmentTemplate.deliverableCities.city.code* | *Code of the city where delivery is made* |
| *shipmentTemplate.deliverableCities.city.name* | *Name of the city where delivery is made* |
| *shipmentTemplate.deliveryFeeType* | [*See: Deliver Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *shipmentTemplate.exchangeAddress.address* | *Return address description* |
| *shipmentTemplate.exchangeAddress.city.code* | *Return address city code* |
| *shipmentTemplate.exchangeAddress.city.name* | *Return address city name* |
| *shipmentTemplate.exchangeAddress.district.id* | *Return address district id* |
| *shipmentTemplate.exchangeAddress.district.name* | *Return address district name* |
| *shipmentTemplate.exchangeAddress.postalCode* | *Return address zip code* |
| *shipmentTemplate.exchangeAddress.title* | *Return address title name* |
| *shipmentTemplate.exchangeInfo* | *Change information/text information will be entered* |
| *shipmentTemplate.installmentInfo* | *Setup information/text information will be entered* |
| *shipmentTemplate.shipmentCompanies.shipmentCompany.name* | *Shipping company name* |
| *shipmentTemplate.shipmentCompanies.shipmentCompany.shortName* | *Shipping company’s short name* |
| *shipmentTemplate.shipmentMethod* | *If the way of reaching the product is 1, the shipment; if 2, the other* |
| *shipmentTemplate.shippingInfo* | *Shipment information/text information will be entered* |
| *shipmentTemplate.specialDelivery* | *Special shipment true/false* |
| *shipmentTemplate.templateName* | *Shipment template* |
| *shipmentTemplate.useDmallCargo* | *Use of n11.com integrated shipping company;*  *1: use*  *0: do not use* |
| *shipmentTemplate.warehouseAddess.title* | *Warehouse address title* |
| *shipmentTemplate.warehouseAddess.address* | *Warehouse address information* |
| *shipmentTemplate.warehouseAddess.district.name* | *Warehouse district name* |
| *shipmentTemplate.warehouseAddess.district.id* | *Warehouse address district id* |
| *shipmentTemplate.warehouseAddess.city.name* | *Warehouse address city name* |
| *shipmentTemplate.warehouseAddess.city.code* | *Warehouse address city code* |
| *shipmentTemplate.warehouseAddess.city.postalCode* | *Warehouse address zip code* |

**CreateOrUpdateShipmentTemplate Sample Call**

**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 Sample Call (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());

}

### Shipment Template Listing (GetShipmentTemplateList)

It is the method used to list the generated shipment template information.

**GetShipmentTemplateListRequest**

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

**\*Bold fields are mandatory fields.**

**GetShipmentTemplateListResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *shipmentTemplates.shipmentTemplate.claimShipmentCompany.name* | *Return/Exchange shipping company’s name* |
| *shipmentTemplates.shipmentTemplate.claimShipmentCompany.shortName* | *Return/Exchange shipping company’s short name* |
| *shipmentTemplates.shipmentTemplate.combinedShipmentAllowed* | [*See: Shipment Allowed*](#_Kargo_Çoklu_Gönderim) |
| *shipmentTemplates.shipmentTemplate.deliverableCities.city.code* | *Code of the city where delivery is made* |
| *shipmentTemplates.shipmentTemplate.deliverableCities.city.name* | *Name of the city where delivery is made* |
| *shipmentTemplates.shipmentTemplate.deliveryFeeType* | [*See: Delivery Fee Type*](#_Kargo_Ödeme_Yöntemi) |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.address* | *Return address description* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.city.code* | *Return address city code* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.city.name* | *Return address city name* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.district.id* | *Return address district id* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.district.name* | *Return address district name* |
| *shipmentTemplates .shipmentTemplate.exchangeAddress.feeCondition* | *Condition type of conditional shipment:*  *1:ByPrice 2: ByUnit* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.feeConditionPrice/feeConditionUnit* | *Condition price/quantity of conditional shipment* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.postalCode* | *Return address zip code* |
| *shipmentTemplates.shipmentTemplate.exchangeAddress.title* | *Return address title name* |
| *shipmentTemplates.shipmentTemplate.exchangeInfo* | *Change information/text information will be entered* |
| *shipmentTemplates.shipmentTemplate.shipmentCompanies.shipmentCompany.name* | *Shipping company name* |
| *shipmentTemplates.shipmentTemplate.shipmentCompanies.shipmentCompany.shortName* | *Shipping company’s short name* |
| *shipmentTemplates.shipmentTemplate.shipmentMethod* | *If the way of reaching the product is 1, the shipment; if 2, the other* |
| *shipmentTemplates.shipmentTemplate.shippingInfo* | *Shipment information/text information will be entered* |
| *shipmentTemplates.shipmentTemplate.specialDelivery* | *Special shipment true/false* |
| *shipmentTemplates.shipmentTemplate.templateName* | *Shipment template* |
| *shipmentTemplates.shipmentTemplate.useDmallCargo* | *Use of n11.com integrated shipping company;*  *1: use*  *0: do not use* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.title* | *Warehouse address title* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.address* | *Warehouse address information* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.district.name* | *Warehouse district name* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.district.id* | *Warehouse address district id* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.city.name* | *Warehouse address city name* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.city.code* | *Warehouse address city code* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.feeCondition* | *Condition type of conditional shipment:*  *1:ByPrice 2: ByUnit* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.feeConditionPrice/feeConditionUnit* | *Condition price/quantity of conditional shipment* |
| *shipmentTemplates.shipmentTemplate.warehouseAddess.city.postalCode* | *Warehouse address zip code* |

**GetShipmentTemplateList Sample Call**

**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 Sample Call (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 Sales/Settlement Service (SettlementService)](#_3.7_Kargolama_Servisi)

WSDL address for Sales/Settlement service: <https://api.n11.com/ws/SettlementService.wsdl>

### [Sales/Settlement Listing (GetSettlementList)](#_Kargo_Şirketlerini_Listeleme)

Service that lists payment information according to the selected date range.

**GetSettlementListRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***startDate*** | *Date range start (YYYY-MM-DD)* |
| ***endDate*** | *Date range end (YYYY-MM-DD)* |
| *pagingData* | [*See: Pagination*](#_Sayfalandırma_(Pagination)_1) |

**GetSettlementListResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *settlementListData.metadata* | [*See: Pagination*](#_Sayfalandırma_(Pagination)_1) |
| *settlementList.deductionAmount* | *Amount of deduction* |
| *settlementList.paymentAmount* | *Amount paid* |
| *settlementList.remittanceDate* | *Remittance date* |
| *settlementList.settlementAmount* | *Settlement amount* |
| *settlementList.settlementDate* | *Settlement date* |
| *settlementList.status* | *Settlement status*  *(“Failed” = Incorrect*  *“NotConfirmed” = Not Confirmed*  *“Confirmed” = Confirmed*  *“Wait\_Remittance” = Waiting for remittance*  *"Success" = Completed)* |

**GetSettlementList Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:GetSettlementListRequest>

<auth>

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

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

</auth>

<startDate>2017-06-01</startDate>

<endDate>2017-07-03</endDate>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

</pagingData>

</sch:GetSettlementListRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

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

<env:Header/>

<env:Body>

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

<result>

<status>success</status>

</result>

<settlementListData>

<metadata>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

<totalCount>32</totalCount>

<pageCount>1</pageCount>

</metadata>

<settlementList>

<deductionAmount>16060.41</deductionAmount>

<paymentAmount>16060.41</paymentAmount>

<remittanceDate>2017-07-04</remittanceDate>

<settlementAmount>0</settlementAmount>

<settlementDate>2017-06-01</settlementDate>

<status>Confirmed</status>

</settlementList>

<settlementList>

<deductionAmount>44363.98</deductionAmount>

<paymentAmount>106534.16</paymentAmount>

<remittanceDate>2017-06-02</remittanceDate>

<settlementAmount>62170.18</settlementAmount>

<settlementDate>2017-06-01</settlementDate>

<status>Success</status>

</settlementList>

...

<settlementList>

<deductionAmount>32109.62</deductionAmount>

<paymentAmount>77293.04</paymentAmount>

<remittanceDate>2017-07-03</remittanceDate>

<settlementAmount>45183.42</settlementAmount>

<settlementDate>2017-07-01</settlementDate>

<status>Wait\_Remittance</status>

</settlementList>

<settlementList>

<deductionAmount>9034.75</deductionAmount>

<paymentAmount>20044.87</paymentAmount>

<remittanceDate>2017-07-03</remittanceDate>

<settlementAmount>11010.12</settlementAmount>

<settlementDate>2017-07-02</settlementDate>

<status>Wait\_Remittance</status>

</settlementList>

</settlementListData>

</ns3:GetSettlementListResponse>

</env:Body>

</env:Envelope>

**GetSettlementList Sample Call (JAVA)**

public static void main(String[] args) {

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

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

String strStartDate = "2017-06-01";

String strEndDate = "2017-07-03";

int currentPage = 0;

int pageSize = 100;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

RequestPagingData pagingData = new RequestPagingData();

pagingData.setCurrentPage(currentPage);

pagingData.setPageSize(pageSize);

GetSettlementListRequest request = new GetSettlementListRequest();

request.setAuth(authentication);

request.setStartDate(strStartDate);

request.setEndDate(strEndDate);

request.setPagingData(pagingData);

SettlementServicePort port = new SettlementServicePortService()

.getSettlementServicePortSoap11();

GetSettlementListResponse response = port.getSettlementList(request);

}

### [Sales/Settlement Detailing (GetSettlementDetail)](#_Kargo_Şirketlerini_Listeleme)

The method showing the details of the payments listed with the **GetSettlementList** method. By selecting a specific day, sales/settlement details for that day are brought.

**GetSettlementDetailRequest**

|  |  |
| --- | --- |
| ***Auth*** | *See:* [*authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***date*** | *Settlement date (YYYY-MM-DD)* |
| *pagingData* | [*See: Pagination*](#_Sayfalandırma_(Pagination)_1) |

**\*Bold fields are mandatory fields.**

**GetSettlementDetailResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *settlementDetail.defermentSettlementUsageDataList.* *defermentSettlementUsageList.amount* | *Order amount* |
| *settlementDetail.defermentSettlementUsageDataList.* *defermentSettlementUsageList.orderDate* | *Order date* |
| *settlementDetail.defermentSettlementUsageDataList.* *defermentSettlementUsageList.orderNumber* | *Order number* |
| *settlementDetail.defermentSettlementUsageDataList.* *defermentSettlementUsageList.productId* | *n11 Product ID* |
| *settlementDetail.defermentSettlementUsageDataList.* *defermentSettlementUsageList.productTitle* | *Product title* |
| *settlementDetail.defermentSettlementUsageDataList.*  *financialTotalDeferment* | *Financial sum of seizure* |
| *settlementDetail.defermentSettlementUsageDataList.*  *totalReturnDefermentAmount* | *Total refund deduction amount* |
| *settlementDetail.orderItemSettlementDataList.metadata* | [*See: Pagination*](#_Sayfalandırma_(Pagination)_1) |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.adFee* | *Advertising amount* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.adFeeVat* | *Advertising VAT amount* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.buyerEmail* | *Buyer e-mail address* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.buyerName* | *Buyer’s name and surname information* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.commission* | *Commission rate* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.completedDate* | *Order completion date* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.deductionAmount* | *Amount of deduction* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.orderItemId* | *Order item ID* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.orderNumber* | *Order number* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.paymentAmount* | *Payment total* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.productId* | *n11 Product ID* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.productName* | *Product title* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.sellerCouponDiscount* | *Seller coupon discount* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.sellerDiscount* | *Seller discount* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.settlementAmount* | *Agreement amount* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.shippingDate* | *Cargo date* |
| *settlementDetail.orderItemSettlementDataList.orderItemSettlementList.trackingNumber* | *Shipment tracking number* |
| *settlementDetail.settlement.deductionAmount* | *Amount of deduction* |
| *settlementDetail.settlement.defermentOfPayment* | *Pending payment* |
| *settlementDetail.settlement.paymentToECash* | *E-money payment* |
| *settlementDetail.settlement.releaseDeferment* | *Released payment* |
| *settlementDetail.settlement.remittance* | *Remittance amount* |
| *settlementDetail.settlement.salesAmount* | *Sales amount* |
| *settlementDetail.settlement.sellerWithdraw* | *Withdrawn seller payment* |

**GetSettlementDetail Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:GetSettlementDetailRequest>

<auth>

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

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

</auth>

<date>2017-06-01</date>

<pagingData>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

</pagingData>

</sch:GetSettlementDetailRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

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

<env:Header/>

<env:Body>

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

<result>

<status>success</status>

</result>

<settlementDetail>

<defermentSettlementUsageDataList>

<defermentSettlementUsageList>

<amount>64.07</amount>

<orderDate>2017-05-18</orderDate>

<orderNumber>201481964698</orderNumber>

<productId>21578970</productId>

<productTitle>Evy Baby Bebek Bezi Kremli Jumbo Paket 42 Adet Maxi Plus 4'lü SET</productTitle>

</defermentSettlementUsageList>

<defermentSettlementUsageList>

<amount>80.74</amount>

<orderDate>2017-04-21</orderDate>

<orderNumber>201369415292</orderNumber>

<productId>123255558</productId>

<productTitle>Prima Bebek Bezi Premium Care 4 Beden Maxi Dev Eko Paket 72 Adet</productTitle>

</defermentSettlementUsageList>

...

<financialTotalDeferment>9088.41</financialTotalDeferment>

<totalReturnDefermentAmount>745.96</totalReturnDefermentAmount>

</defermentSettlementUsageDataList>

<orderItemSettlementDataList>

<metadata>

<currentPage>0</currentPage>

<pageSize>100</pageSize>

<totalCount>18</totalCount>

<pageCount>1</pageCount>

</metadata>

<orderItemSettlementList>

<adFee>0.09</adFee>

<adFeeVat>0.02</adFeeVat>

<buyerEmail>\*\*\*\*\*\*@h\*\*\*\*\*.com</buyerEmail>

<buyerName>E\*\*\*N Y\*\*\*\*N</buyerName>

<commission>8</commission>

<completedDate>2017-06-01</completedDate>

<deductionAmount>12.62</deductionAmount>

<orderItemId>58061926</orderItemId>

<orderNumber>202741342695</orderNumber>

<paymentAmount>29.9</paymentAmount>

<paymentDate>2017-05-28</paymentDate>

<productId>184730259</productId>

<productName>Colgate Optik Beyaz Ağız Bakım Seti Çanta &amp; Saklama Kabı Hediyeli</productName>

<sellerCouponDiscount>0</sellerCouponDiscount>

<sellerDiscount>11</sellerDiscount>

<settlementAmount>17.28</settlementAmount>

<shippingDate>2017-05-29</shippingDate>

<trackingNumber>100250046277</trackingNumber>

</orderItemSettlementList>

...

</orderItemSettlementDataList>

<settlement>

<deductionAmount>60424.39</deductionAmount>

<defermentOfPayment>745.96</defermentOfPayment>

<paymentToECash>0</paymentToECash>

<releaseDeferment>0</releaseDeferment>

<remittance>62170.18</remittance>

<salesAmount>122594.57</salesAmount>

<sellerWithdraw>0</sellerWithdraw>

</settlement>

</settlementDetail>

</ns3:GetSettlementDetailResponse>

</env:Body>

</env:Envelope>

**GetSettlementDetail Sample Call (JAVA)**

public static void main(String[] args) {

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

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

String strDate = "2017-06-01";

int currentPage = 0;

int pageSize = 100;

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

RequestPagingData pagingData = new RequestPagingData();

pagingData.setCurrentPage(currentPage);

pagingData.setPageSize(pageSize);

GetSettlementDetailRequest request = new GetSettlementDetailRequest();

request.setAuth(authentication);

request.setDate(strDate);

request.setPagingData(pagingData);

SettlementServicePort port = new SettlementServicePortService()

.getSettlementServicePortSoap11();

GetSettlementDetailResponse response = port.getSettlementDetail(request);

}

## 3.10 Store Request Service (TicketService)

WSDL address for Store Request service: [https://api.n11.com/ws/TicketService.wsdl](https://api.n11.com/ws/SettlementService.wsdl)

### Listing the Requests Assigned to the Store (TicketListingAssignedToSeller)

Service that lists the requests assigned to the store in a list.

**TicketListingAssignedToSellerRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***sellerId*** | *Store ID* |
| ***first*** | The list shows the order in which the tickets will start. |
| ***pageSize*** | *Number of items on the page* |

**TicketListingAssignedToSellerResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *totalNumberOfTickets* | *Total number of requests assigned to the store* |
| *ticketList.* *ticketId* | *Request ID* |
| *ticketList.* *header* | *Request Title* |
| *ticketList.* *content* | *Request Content* |
| *ticketList.* *createDate* | *Request creation date* |
| *ticketList.* *ticketStatus* | *Request Status*  *(OPENED=Opened,*  *CLOSED=CLOSED,*  *WAITING=WAITING*  *)* |
| *ticketList.* *ticketPriorStatus* | *Request Severity Status*  *(HIGH\_PRIOR = Very Important,*  *PRIOR=Important,*  *STANDARD = Standard*  *)* |
| *ticketList.* *isReadBySeller* | *It keeps the information of the request read or unread. It can be True or False.* |
| *ticketList.creator.email* | *E-mail of the creator of the request* |
| ticketList.creator.id | *The id of the creator of the request* |
| ticketList.creator.name | *Name of the creator of the request* |
| ticketList.creator.phone | *Number of the creator of the request* |
| ticketList.reason.id | Id of reason for creating the request |
| ticketList.reason.name | Reason for creating the request |
| ticketList.files.file.fileName | The file name of the request |
| ticketList.files.file.fileUrl | Download url of the request file |
| ticketList. answers.answer.id | The id of the response to the request |
| ticketList.answers.answer.content | The content of the response to the request |
| ticketList.answers.answer.createDate | The date of the response to the request |
| ticketList.answers.answer.isReadBySeller | *It keeps the information of whether the response is read or unread. It can be True or False.* |
| ticketList.answers.answer.creator.email | E-mail of the person responding to the request |
| ticketList.answers.answer.creator.id | ID of the person responding to the request |
| ticketList.answers.answer.creator.name | Name of the person responding to the request |
| ticketList.answers.answer.creator.phone | Telephone number of the person responding to the request |
| ticketList. answers.answer.files.file.fileName | Name of the response file |
| ticketList. answers.answer.files.file.fileUrl | Download url of the response file |
| ticketList.seller.email | Mail of the seller to whom the request was assigned |
| ticketList.seller.id | ID of the seller to whom the request was assigned |
| ticketList.seller.name | Name of the seller to whom the request was assigned |
| ticketList.seller.nickName | Alias of the seller to whom the request was assigned |
| ticketList.seller.phone | Telephone number of the seller to whom the request was assigned |
|  |  |

**TicketListingAssignedToSeller Sample Call**

**Request**

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

<soapenv:Header />

<soapenv:Body>

<sch:TicketListingAssignedToSellerRequest>

<auth>

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

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

</auth>

<sellerId>13050</sellerId>

<first>0</first>

<pageSize>2</pageSize>

</sch:TicketListingAssignedToSellerRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

<ticketList>

<ticket>

<ticketId>20053</ticketId>

<header>header</header>

<content>content</content>

<createDate>2019-10-22T08:53:53.828+03:00</createDate>

<ticketStatus>OPENED</ticketStatus>

<ticketPriorStatus>PRIOR</ticketPriorStatus>

<isReadBySeller>false</isReadBySeller>

<creator>

<email>admin@dmall.com</email>

<id>21</id>

<name>admin</name>

<phone>5327263183</phone>

</creator>

<reason>

<id>2059</id>

<name>Modify Member Information</name>

</reason>

<files>

<file>

<fileName>Book1 copy 4.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketFile\_20053/1571730725974\_Book1 copy 4.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy 3.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketFile\_20053/1571730724873\_Book1 copy 3.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy 2.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketFile\_20053/1571730723446\_Book1 copy 2.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/templateFile\_7250/1565547000382\_Book1 copy.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy 6.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/templateFile\_7250/1565546998857\_Book1 copy 6.xlsx</fileUrl>

</file>

</files>

<answers>

<answer>

<id>21050</id>

<content>content</content>

<createDate>2019-10-22T10:36:47.207+03:00</createDate>

<creator>

<email>admin@dmall.com</email>

<id>21</id>

<name>admin</name>

<phone>\*\*\*\*</phone>

</creator>

<isReadBySeller>false</isReadBySeller>

<files>

<file>

<fileName>Book1.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketAnswerFile\_21050/1571729822266\_Book1.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy 3.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketAnswerFile\_21050/1571729818980\_Book1 copy 3.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy 5.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketAnswerFile\_21050/1571729820345\_Book1 copy 5.xlsx</fileUrl>

</file>

</files>

</answer>

<answer>

<id>21100</id>

<content>content</content>

<createDate>2019-10-22T10:38:11.666+03:00</createDate>

<creator>

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

<id>13050</id>

<name>seller</name>

<nickName>sellerRumuz</nickName>

<phone>4444444444</phone>

</creator>

<isReadBySeller>false</isReadBySeller>

<files>

<file>

<fileName>Book1 copy 4.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketAnswerFile\_21100/1571729925463\_Book1 copy 4.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy 3.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketAnswerFile\_21100/1571729925454\_Book1 copy 3.xlsx</fileUrl>

</file>

</files>

</answer>

</answers>

</ticket>

<ticket>

<ticketId>20052</ticketId>

<header>header</header>

<content>content</content>

<createDate>2019-10-22T08:51:49.996+03:00</createDate>

<ticketStatus>OPENED</ticketStatus>

<ticketPriorStatus>STANDART</ticketPriorStatus>

<isReadBySeller>false</isReadBySeller>

<creator>

<email>admin@dmall.com</email>

<id>21</id>

<name>admin</name>

<phone>\*\*\*\*</phone>

</creator>

<reason>

<id>2059</id>

<name>Modify Member Information</name>

</reason>

<files>

<file>

<fileName>Book1 copy 4.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketFile\_20052/1571723538458\_Book1 copy 4.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy 5.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketFile\_20052/1571723539426\_Book1 copy 5.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy 3.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketFile\_20052/1571723536950\_Book1 copy 3.xlsx</fileUrl>

</file>

</files>

<answers/>

</ticket>

</ticketList>

<totalNumberOfTickets>16</totalNumberOfTickets>

<seller>

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

<id>13050</id>

<name>seller</name>

<nickName>sellerRumuz</nickName>

<phone>4444444444</phone>

</seller>

</ns3:TicketListingResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**TicketListingAssignedToSeller Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
**int** first=10;  
**int** pageSize=10;  
String sellerId=123;  
  
Authentication authentication = **new** Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
  
TicketListingAssignedToSellerSARequest request = **new** TicketListingAssignedToSellerSARequest();  
request.setAuthentication(authentication);  
request.setFirst(first);  
request.setPageSize(pageSize);  
request.setSellerId(sellerId);  
  
TicketServicePort port = **new** TicketServicePortService()  
 .getTicketServicePortPortSoap11();  
SellerTicketListingAssignToSAResponse response = port.listTicketAssignedToSeller(request);

}

### Listing the Requests Opened by the Store (TicketListingBelongsToSeller)

The service that sends the requests opened by the store in a list.

**TicketListingBelongsToSellerRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***sellerId*** | *Store ID* |
| ***first*** | The list shows the order in which the tickets will start. |
| ***pageSize*** | *Number of items on the page* |

**TicketListingBelongsToSellerResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *totalNumberOfTickets* | *Total number of requests opened by the store* |
| *ticketList.* *ticketId* | *Request ID* |
| *ticketList.* *header* | *Request Title* |
| *ticketList.* *content* | *Request Content* |
| *ticketList.* *createDate* | *Request creation date* |
| *ticketList.* *ticketStatus* | *Request Status*  *(OPENED=Opened,*  *CLOSED=CLOSED,*  *WAITING=WAITING*  *)* |
| *ticketList.* *ticketPriorStatus* | *Request Severity Status*  *(HIGH\_PRIOR = Very Important,*  *PRIOR=Important,*  *STANDARD = Standard*  *)* |
| *ticketList.* *isReadBySeller* | *The information on whether the request has been read or not can be True or False.* |
| *ticketList.reason.id* | *Id of reason for creating the request* |
| ticketList.reason.name | *Reason for creating the request* |
| ticketList.files.file.fileName | *The file name of the request* |
| ticketList.files.file.fileUrl | *Download url of the request file* |
| ticketList. answers.answer.id | The id of the response to the request |
| ticketList.answers.answer.content | The content of the response to the request |
| ticketList.answers.answer.createDate | The date of the response to the request |
| ticketList.answers.answer.isReadBySeller | *The information on whether the response has been read or not can be True or False.* |
| ticketList.answers.answer.creator.email | E-mail of the person responding to the request |
| ticketList.answers.answer.creator.id | ID of the person responding to the request |
| ticketList.answers.answer.creator.name | Name of the person responding to the request |
| ticketList.answers.answer.creator.phone | Telephone number of the person responding to the request |
| ticketList. answers.answer.files.file.fileName | Name of the response file |
| ticketList. answers.answer.files.file.fileUrl | Download url of the response file |
| ticketList.creator.email | E-mail of the store opening your request |
| ticketList. creator.id | ID of the store opening the request |
| ticketList. creator.name | Name of the store opening the request |
| ticketList. creator.nickName | Alias of the store opening the request |
| ticketList. creator.phone | Telephone number of the store opening the request |

**TicketListingBelongsToSeller Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:TicketListingBelongsToSellerRequest>

<auth>

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

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

</auth>

<sellerId>13050</sellerId>

<first>0</first>

<pageSize>10</pageSize>

</sch:TicketListingBelongsToSellerRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

<ticketList>

<ticket>

<ticketId>18050</ticketId>

<header>header</header>

<content>content</content>

<createDate>2019-10-21T10:58:43.978+03:00</createDate>

<ticketStatus>OPENED</ticketStatus>

<ticketPriorStatus>STANDART</ticketPriorStatus>

<isReadBySeller>true</isReadBySeller>

<reason>

<id>2063</id>

<name>BKM</name>

</reason>

<files>

<file>

<fileName>Book1 copy 4.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketFile\_18050/1571999649546\_Book1 copy 4.xlsx</fileUrl>

</file>

<file>

<fileName>Book1 copy 3.xlsx</fileName>

<fileUrl>https://so.n11.com/sellerTicket/ticketFile\_18050/1571999649544\_Book1 copy 3.xlsx</fileUrl>

</file>

</files>

<answers>

<answer>

<id>20050</id>

<content>content</content>

<createDate>2019-10-21T10:58:58.685+03:00</createDate>

<creator>

<email>admin@dmall.com</email>

<id>21</id>

<name>admin</name>

<phone>\*\*\*\*</phone>

</creator>

<isReadBySeller>false</isReadBySeller>

<files/>

</answer>

<answer>

<id>20051</id>

<content>content</content>

<createDate>2019-10-21T11:02:32.759+03:00</createDate>

<creator>

<email>admin@dmall.com</email>

<id>21</id>

<name>admin</name>

<phone>5327263183</phone>

</creator>

<isReadBySeller>false</isReadBySeller>

<files/>

</answer>

<answer>

<id>20052</id>

<content>content<content>

<createDate>2019-10-21T13:45:23.316+03:00</createDate>

<creator>

<email>admin@dmall.com</email>

<id>21</id>

<name>admin</name>

<phone>5327263183</phone>

</creator>

<isReadBySeller>false</isReadBySeller>

<files/>

</answer>

<answer>

<id>20055</id>

<content>content</content>

<createDate>2019-10-21T14:09:48.835+03:00</createDate>

<creator>

<email>admin@dmall.com</email>

<id>21</id>

<name>admin</name>

<phone>5327263183</phone>

</creator>

<isReadBySeller>false</isReadBySeller>

<files/>

</answer>

<answer>

<id>20056</id>

<content>content</content>

<createDate>2019-10-21T14:10:13.683+03:00</createDate>

<creator>

<email>admin@dmall.com</email>

<id>21</id>

<name>admin</name>

<phone>5327263183</phone>

</creator>

<isReadBySeller>false</isReadBySeller>

<files/>

</answer>

<answer>

<id>20057</id>

<content>content</content>

<createDate>2019-10-21T14:10:52.259+03:00</createDate>

<creator>

<email>admin@dmall.com</email>

<id>21</id>

<name>admin</name>

<phone>\*\*\*\*</phone>

</creator>

<isReadBySeller>false</isReadBySeller>

<files/>

</answer>

</answers>

</ticket>

</ticketList>

<totalNumberOfTickets>1</totalNumberOfTickets>

<creator>

<email>tunc.dogan@n11.com</email>

<id>13050</id>

<name>seller</name>

<nickName>sellerRumuz</nickName>

<phone>4444444444</phone>

</creator>

</ns3:TicketListingResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**TicketListingBelongsToSeller Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
**int** first=10;  
**int** pageSize=10;  
String sellerId=123;  
  
Authentication authentication = **new** Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
  
TicketListingBelongsToSellerSARequest

request = **new** TicketListingBelongsToSellerSARequest();  
request.setAuthentication(authentication);  
request.setFirst(first);  
request.setPageSize(pageSize);  
request.setSellerId(sellerId);  
  
TicketServicePort port = **new** TicketServicePortService()  
 .getTicketServicePortPortSoap11();  
SellerTicketListingBelongsToSAResponse

response = port. listTicketBelongsToSeller(request);

}

### Store's Response to Requests (TicketAnswer)

A service that enables the store to respond to opened requests or the requests opened by itself.

**TicketAnswerRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***sellerId*** | *Responding store Id* |
| ***ticketId*** | Responded request Id |
| ***content*** | *Response Content* |

**TicketAnswerResponse**

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

**TicketAnswer Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:TicketAnswerRequest>

<auth>

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

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

</auth>

<sellerId>13050</sellerId>

<ticketId>20052</ticketId>

<content>seller api cevap</content>

</sch:TicketAnswerRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

</ns3:TicketAnswerResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**TicketAnswer Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
String content=**"content"**;  
**long** sellerId=123;  
**long** ticketId=12345;  
  
Authentication authentication = **new** Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
  
SellerTicketAnswerSaRequest request = **new** SellerTicketAnswerSaRequest();  
request.setAuthentication(authentication);  
request.setContent(content);  
request.setSellerId(sellerId);  
request.setTicketId(ticketId);  
  
TicketServicePort port = **new** TicketServicePortService()  
 .getTicketServicePortPortSoap11();  
SellerTicketAnswerSAResponse response = port. answerTicket

(request);

}

### Request Opening of the Store (TicketCreate)

The service that enables the store to open request.

**TicketCreateRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***reasonId*** | *Id of request opening reason* |
| ***sellerId*** | Id of the store that opened the request |
| ***header*** | *Request Title* |
| **content** | Request Content |

**TicketCreateResponse**

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

**TicketCreate Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:TicketCreateRequest>

<auth>

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

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

</auth>

<reasonId>2056</reasonId>

<sellerId>13050</sellerId>

<header>header</header>

<content>content</content>

</sch:TicketCreateRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

</ns3:TicketCreateResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**TicketCreate Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
String content = **"content"**;  
String header = **"header"**;  
**long** reasonId = 12;  
**long** sellerId = 123;  
**long** ticketId = 12345;  
  
Authentication authentication = **new** Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
  
SellerTicketCreateSaRequest request = **new** SellerTicketCreateSaRequest();  
request.setAuthentication(authentication);  
request.setContent(content);  
request.setSellerId(sellerId);  
request.setHeader(header);  
request.setReasonId(reasonId);  
  
  
TicketServicePort port = **new** TicketServicePortService()  
 .getTicketServicePortPortSoap11();  
SellerTicketCreateSAResponse response = port.createTicket(request);

}

### Store Request Reasons (TicketReasons)

Service that conveys the information of the reasons to enter when opening a store request.

The data sent includes the Root-Parent-Child-Child relationship.

ReasondId must send the child id in the 3rd level when store opening the request.

The isReqired field exists only in 4.level reason. It can be True or False.

**TicketReasonsRequest**

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

**TicketReasonsResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *id* | *Id of the reason for opening an incoming request* |
| *name* | *Name of the reason for opening an incoming request* |

**TicketReasons Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:TicketReasonsRequest>

<auth>

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

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

</auth>

</sch:TicketReasonsRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

<allTicketReasons>

<reason>

<id>9051</id>

<name>1st</name>

<children>

<id>9052</id>

<name>2nd</name>

</children>

</reason>

<reason>

<id>11100</id>

<name>3st</name>

<children>

<id>11101</id>

<name>2st</name>

<children>

<id>11102</id>

<name>1st</name>

<children>

<id>11103</id>

<name>0st</name>

<isRequired>false</isRequired>

</children>

<children>

<id>11104</id>

<name>-1st</name>

<isRequired>true</isRequired>

</children>

<children>

<id>11105</id>

<name>-2st</name>

<isRequired>true</isRequired>

</children>

<children>

<id>11107</id>

<name>aa</name>

<isRequired>true</isRequired>

</children>

<children>

<id>12050</id>

<name>-4</name>

<isRequired>false</isRequired>

</children>

</children>

</children>

</reason>

<reason>

<id>2050</id>

<name>Buyer child 12</name>

<children>

<id>2052</id>

<name>Member</name>

<children>

<id>2057</id>

<name>Registration</name>

</children>

<children>

<id>2058</id>

<name>Withdraw</name>

</children>

<children>

<id>2059</id>

<name>Modify Member Information</name>

</children>

<children>

<id>2060</id>

<name>Login</name>

<children>

<id>10050</id>

<name>loginsub</name>

<isRequired>true</isRequired>

</children>

<children>

<id>10051</id>

<name>loginsub2</name>

<isRequired>true</isRequired>

</children>

</children>

</children>

<children>

<id>2053</id>

<name>Order / Payment</name>

<children>

<id>2061</id>

<name>Order</name>

</children>

<children>

<id>2062</id>

<name>Card Payment</name>

</children>

<children>

<id>2063</id>

<name>BKM</name>

<children>

<id>10052</id>

<name>bkm2</name>

<isRequired>false</isRequired>

</children>

</children>

<children>

<id>2064</id>

<name>Bonus Pay</name>

</children>

</children>

<children>

<id>2054</id>

<name>Shipment</name>

<children>

<id>2065</id>

<name>Shipping Fee</name>

</children>

<children>

<id>2066</id>

<name>Shipping Tracking</name>

</children>

<children>

<id>2067</id>

<name>Order Confirm</name>

<children>

<id>11050</id>

<name>kk</name>

<isRequired>false</isRequired>

</children>

</children>

</children>

<children>

<id>2055</id>

<name>etc</name>

<children>

<id>2056</id>

<name>How to use dMall</name>

</children>

</children>

<children>

<id>9053</id>

<name>test 2nd</name>

</children>

<children>

<id>9054</id>

<name>2nd v2</name>

</children>

</reason>

<reason>

<id>5051</id>

<name>Call Center</name>

<children>

<id>5052</id>

<name>Call Center</name>

</children>

</reason>

<reason>

<id>2051</id>

<name>Seller</name>

<children>

<id>2068</id>

<name>Member</name>

<children>

<id>2074</id>

<name>Registration</name>

</children>

<children>

<id>2075</id>

<name>Modify Member Information</name>

</children>

</children>

<children>

<id>2069</id>

<name>Product</name>

<children>

<id>2076</id>

<name>Product Registration</name>

</children>

<children>

<id>2077</id>

<name>Product Info Mgmt</name>

</children>

</children>

<children>

<id>2070</id>

<name>AD</name>

<children>

<id>2071</id>

<name>AD Type</name>

</children>

<children>

<id>2072</id>

<name>How to apply AD</name>

</children>

<children>

<id>2073</id>

<name>AD Mgmt</name>

</children>

</children>

<children>

<id>8050</id>

<name>wqe</name>

</children>

</reason>

<reason>

<id>9050</id>

<name>dsffdsdfs</name>

</reason>

</allTicketReasons>

</ns3:TicketReasonsResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**TicketReasons Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
  
Authentication authentication = **new** Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
  
SellerTicketReasonsSARequest request = **new** SellerTicketReasonsSARequest();  
request.setAuthentication(authentication);

TicketServicePort port = **new** TicketServicePortService()  
 .getTicketServicePortPortSoap11();

SellerTicketReasonsSAResponse ticketReasonsResponse = port.generateReasons

(request);

### Request and Response Reading (TicketRead)

Service that updates the status of read requests and responses as read

When a request is made with SellerId and TicketId, it updates the request and all its responses as read.

TicketListingAssignedToSeller and TicketListingBelongsToSeller

It can be seen by checking the isReadBySeller field that belongs to ticket and response in their service.

**TicketReadRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***ticketId*** | The id of the request whose status is updated. |
| ***sellerId*** | ID of the store that reads the request and the response |

**TicketReasonsResponse**

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

**TicketRead Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:TicketReadRequest>

<auth>

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

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

</auth>

<ticketId>30100</ticketId>

<sellerId>522</sellerId>

</sch:TicketReadRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

</ns3:TicketReadResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**TicketRead Sample Call (JAVA)**

public static void main(String[] args) {

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

long ticketId=123;

long sellerId=1234  
  
Authentication authentication = **new** Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
  
SellerTicketReadRequest request = **new** SellerTicketReadRequest();  
request.setAuthentication(authentication);

request.setTicketId(ticketId)

request.setSellerId(sellerId)

TicketServicePort port = **new** TicketServicePortService()  
 .getTicketServicePortPortSoap11();

SellerTicketReadResponse ticketReadSAResponse = port. readTicket(request);

## 3.11 Sap Commission E-Invoice Detail Service (SapCommissionEInvoiceDetailService)

Sap Commission E-Invoice Detail Service WSDL address: <https://api.n11.com/ws/SapCommissionEInvoiceDetailService.wsdl>

### Invoice Detailing (GetSapCommissionEInvoiceDetail)

The daily query limit number for the invoice detail is set to 3.

**GetSapCommissionEInvoiceDetailRequest**

|  |  |
| --- | --- |
| ***Auth*** | *See:* [*authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***date*** | *Date to be received invoice detail (YYYY-MM)* |

**GetSapCommissionEInvoiceDetailResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *addOrder* | *Advertisement amount* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *belnr* | *Document Number* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *malzemeKodu* | *Material Code* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *miktar* | *Quantity* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *orderItemId* | *Order Item Id* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *orderNumber* | *Order Number* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *pazarlamaGeliri* | *Marketing Revenue* |
| commissionInvoiceDetail. commissionInvoiceDetailDataList. commissionInvoiceItemDetailData. saticiId | *Seller Id* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *tamamlanmaTarihi* | *Completion Date* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *transactionDDText* | *Sales Type Definition* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *transactionType* | *Sales Type* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *tutar* | *Amount* |
| *commissionInvoiceDetail.* *commissionInvoiceDetailDataList.* *commissionInvoiceItemDetailData.* *vadeFarki* | *Maturity Difference* |

**GetSapCommissionEInvoiceDetail Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch: GetSapCommissionEInvoiceDetailRequest >

<auth>

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

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

</auth>

<date>2017-06</date>

</sch: GetSapCommissionEInvoiceDetailRequest >

</soapenv:Body>

</soapenv:Envelope>

**Response**

<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header />

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

<commissionInvoiceDetail>

<commissionInvoiceDetailDataList>

<commissionInvoiceItemDetailData>

<addOrder>0,00</addOrder>

<belnr>0400160916</belnr>

<malzemeKodu />

<miktar>1</miktar>

<orderItemId>37209469</orderItemId>

<orderNumber>207534669362</orderNumber>

<pazarlamaGeliri>0,00</pazarlamaGeliri>

<saticiId />

<tamamlanmaTarihi>03.08.2016</tamamlanmaTarihi>

<transactionDDText>Satış Kargo (POP)</transactionDDText>

<transactionType>SKP</transactionType>

<tutar>4,25</tutar>

<vadeFarki>0,00</vadeFarki>

</commissionInvoiceItemDetailData>

</commissionInvoiceDetailDataList>

</commissionInvoiceDetail>

</ns3:GetSapCommissionEInvoiceDetailResponse> </SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**Response errorMessage descriptions**

| *errorMessage* | *Description* |
| --- | --- |
| *USER\_NOT\_FOUND* | *Indicates the situation in which the relevant seller was not found* |
| *NO\_DATA\_FOUND* | *Indicates the situation where the invoice details cannot be found for the entered date* |
| *InvalidDate* | *Indicates that an invalid date format is used* |
| *INVALID\_RESPONSE\_CODE* | *Indicates the status in which the invoice detail  could not be received* |
| *dailyEInvoiceSearchLimit* | *Indicates that the daily query limit has been reached* |

**GetSapCommissionEInvoiceDetail Sample Call (JAVA)**

public static void main(String[] args) {

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

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

String date = "2019-06";

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

GetSapCommissionEInvoiceDetailRequest request = new GetSapCommissionEInvoiceDetailRequest ();

request.setAuth(authentication);

request.setDate(date);

SapCommissionInvoiceDetailServicePort port = new SapCommissionInvoiceDetailServicePortService()

.getSapCommissionInvoiceDetailServicePortSoap11();

GetSettlementDetailResponse response = port.getSapCommissionInvoiceDetail(request);}

## 3.12 Sap Bank Statement E-Invoice Service (SapBankStatementEInvoiceService)

Sap Bank Statement E-Invoice Service WSDL address:

[https://api.n11.com/ws/ SapBankStatementEInvoiceService.wsdl](https://api.n11.com/ws/%20SapBankStatementEInvoiceService.wsdl)

### Bank Statement (GetSapBankStatementEInvoice)

The number of daily inquiry limits for bank statement is set to 3.

**GetSapBankStatementEInvoiceRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***startDate*** | *Date range start (YYYY-MM-DD)* |
| ***endDate*** | *Date range end (YYYY-MM-DD)* |

**GetSapBankStatementEInvoiceResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.ALACAK* | *Credit* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.BELNR* | *Document Number* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.BLART* | *Document Type* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.BLART\_LTEXT* | *Document Type Definition* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.BORC* | *Debt* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.BUDAT* | *Date of registration* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.KONTO* | *Account* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.NBAKIYE* | *Net Balance* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.SGTXT* | *Text* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData* *ZUONR* | *Designation* |
| *bankStatementInvoiceDataList.*  *bankStatementInvoiceItemData.WAERS* | *Document Currency* |

**GetSapBankStatementEInvoice Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:GetSapBankStatementEInvoiceRequest>

<auth>

<appKey>030c661a-2e92-4084-a94e-b856917c9781</appKey>

<appSecret>123qwe</appSecret>

</auth>

<startDate>2015-01-01</startDate>

<endDate>2020-01-01</endDate>

</sch:GetSapBankStatementEInvoiceRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

<bankStatementInvoiceDataList>

<bankStatementInvoiceItemData>

<ALACAK>25,00</ALACAK>

<BELNR>0100001790</BELNR>

<BLART>SA</BLART>

<BLART\_LTEXT>Ana hesap belgesi</BLART\_LTEXT>

<BORC>0,00</BORC>

<BUDAT>30.03.2018</BUDAT>

<KONTO>102208</KONTO>

<NBAKIYE>0,00</NBAKIYE>

<SGTXT>DENKLEŞTİRME VIRMANI</SGTXT>

<WAERS>TRY</WAERS>

<ZUONR>0000102208</ZUONR>

</bankStatementInvoiceItemData>

<bankStatementInvoiceItemData>

<ALACAK>0,00</ALACAK>

<BELNR>0100005237</BELNR>

<BLART>AB</BLART>

<BLART\_LTEXT>Muhasebe belgesi</BLART\_LTEXT>

<BORC>0,00</BORC>

<BUDAT>30.03.2018</BUDAT>

<KONTO>102208</KONTO>

<NBAKIYE>0,00</NBAKIYE>

<SGTXT/>

<WAERS>TRY</WAERS>

<ZUONR/>

</bankStatementInvoiceItemData>

</bankStatementInvoiceDataList>

</ns3:GetSapBankStatementEInvoiceResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**Response errorMessage descriptions**

| *errorMessage* | *Description* |
| --- | --- |
| *SELLER\_NOT\_FOUND* | *Indicates the situation in which the relevant seller was not found* |
| *NO\_DATA\_FOUND* | *Indicates the situation where the bank statement could not be found for the entered date* |
| *InvalidDate* | *Indicates that an invalid date format is used* |
| *INTERNAL\_ERROR* | *Indicates the situation where the bank statement could not be obtained* |
| *dailyBankStatementEInvoiceSearchLimit* | *Indicates that the daily query limit has been reached* |

**GetSapBankStatementEInvoice Sample Call (JAVA)**

public static void main(String[] args) {

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

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

String startDate = "2019-06-01";

String endDate = "2019-07-01";

Authentication authentication = new Authentication();

authentication.setAppKey(strAppKey);

authentication.setAppSecret(strAppSecret);

GetSapBankStatementEInvoiceRequest request = new GetSapBankStatementEInvoiceRequest ();

request.setAuth(authentication);

request.setStartDateDate(date);

request.setEndDate(date);

SapBankStatementInvoiceServicePort port = new SapBankStatementEnvoiceServicePortService()

.getSapSapBankStatementEnvoiceServicePortSoap11();

GetBankStatementEInvoiceResponse response = port.getSapBankStatementInvoice(request);}

## 3.13 Order Cancellation Requests Service (ClaimCancelService)

WSDL address for Order Cancellation Requests service:

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

### Listing the Cancellation Requests (ClaimCancelList)

The service that sends the order cancellation requests to the store as a list. 20 cancellation information is displayed on each page.

**ClaimCancelListRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| *searchData.status* | [*See: ClaimCancelStatus*](#_Sipariş_İptal_Talebi) |
| *searchData.executer* | [*See: ClaimExecuterType*](#_İşlemi_Yapan_(Claim) |
| *searchData.searchInfoType* | [*See: ClaimSearchType*](#_İptal_Talebi_Detayı) |
| *searchData.searchQuery* | *The word / code to search by searchData.searchInfoType.* |
| *searchData.searchDate.searchDateType* | [*See: ClaimCancelDateType*](#_İptal_Talebi_Arama) |
| *searchData.searchDate.period.startDate* | *Start date by searchDateType.* *(gg/aa/yyyy)* |
| *searchData.searchDate.period.endDate* | *End date by searchDateType.* *(gg/aa/yyyy)* |
| *currentPage* | *The page number shown (currentpage starts with 0).* |

**\*Bold fields are mandatory fields.**

**ClaimCancelListResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *claimCancelList.claimCancel.claimCancelId* | *Cancellation request id.* |
| *claimCancelList.claimCancel.status* | *Cancellation status.* |
| *claimCancelList.claimCancel.executer* | *The information about the person making the transaction* |
| *claimCancelList.claimCancel.cancelReasonType* | *Reason for cancellation.* |
| *claimCancelList.claimCancel.cancelReasonDescription* | *Cancellation description.* |
| *claimCancelList.claimCancel.orderNumber* | *Order code.* |
| *claimCancelList.claimCancel.shipmentMethod* | *Shipment Method.* |
| *claimCancelList.claimCancel.bundleName* | *Special Offer Name (If any).* |
| *claimCancelList.claimCancel.requestDate* | *Date of request.* |
| *claimCancelList.claimCancel.deniedDate* | *Rejection Date (If the cancellation status is 'Denied')* |
| *claimCancelList.claimCancel.completedDate* | *Acceptance Date (If the cancellation status is 'Completed').* |
| *claimCancelList.claimCancel.productId* | *Product Code.* |
| *claimCancelList.claimCancel.productName* | *Product Name.* |
| *claimCancelList.claimCancel.quantity* | *Number of products with cancellation requests.* |
| *claimCancelList.claimCancel.unitPrice* | *Unit Price.* |
| *claimCancelList.claimCancel.finalPrice* | *Amount Paid.* |
| *claimCancelList.claimCancel.denyReasonType* | *Reason for Rejection (If the cancellation status is 'Denied').* |
| *claimCancelList.claimCancel.denyReasonDescription* | *Rejection Description (If the cancellation status is 'Denied').* |
| *claimCancelList.claimCancel.shipmentCompany* | *Shipping Company.* |
| *claimCancelList.claimCancel.deliveryFeeType* | *Information about who paid the Shipping Fee;*  *'ByBuyer' Buyer Pays*  *'BySeller' Store Pays*  *‘Conditional’ Conditional Shipment* |
| *claimCancelList.claimCancel.buyerPhone* | *Customer’s Phone Number.* |
| *claimCancelList.claimCancel.buyerEmail* | *Customer’s E-mail Address.* |
| *claimCancelList.claimCancel.buyerName* | *Customer’s Name & Surname.* |
| *claimCancelList.claimCancel.paymentDate* | *Payment Date.* |
| *pagingData* | [*See: Pagination*](#_Sayfalandırma_(Pagination)_2) |

**ClaimCancelList Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:ClaimCancelListRequest>

<auth>

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

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

</auth>

<searchData>

<status>REQUESTED</status>

<executer>BUYER</executer>

<searchInfoType>ORDERID</searchInfoType>

<searchQuery>123456789</searchQuery>

<searchDate>

<searchDateType>REQUESTED</searchDateType>

<period>

<startDate>01/01/2019</startDate>

<endDate>30/12/2019</endDate>

</period>

</searchDate>

</searchData>

<pagingData>

<currentPage>0</currentPage>

</pagingData>

</sch:ClaimCancelListRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

<claimCancelList>

<claimCancel>

<buyerEmail>\*\*\*@gmail.com</buyerEmail>

<buyerName>name surname</buyerName>

<buyerPhone>\*\*\*\*</buyerPhone>

<cancelReasonDescription>trial description</cancelReasonDescription>

<cancelReasonType>Cancel All</cancelReasonType>

<claimCancelId>1234</claimCancelId>

<deliveryFeeType>ByBuyer</deliveryFeeType>

<executer>BUYER</executer>

<finalPrice>543</finalPrice>

<orderNumber>123456789</orderNumber>

<paymentDate>18/11/2019</paymentDate>

<productId>3456</productId>

<productName>trialproduct3</productName>

<quantity>1</quantity>

<requestDate>18/11/2019</requestDate>

<shipmentCompany>Yurtiçi</shipmentCompany>

<shipmentMethod>Shipping</shipmentMethod>

<status>REQUESTED</status>

<unitPrice>543</unitPrice>

</claimCancel>

</claimCancelList>

<pagingData>

<currentPage>0</currentPage>

<pageSize>20</pageSize>

<totalCount>1</totalCount>

<pageCount>1</pageCount>

</pagingData>

</ns3:ClaimCancelListResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimCancelList Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
String status =**"REQUESTED"**;  
String executer =**"BUYER"**;

String searchInfoType =**"ORDERID"**;

String searchQuery =**"123456789"**;

String searchDateType=**"REQUESTED"**;

String startDate = "01/01/2019";

String endDate = "30/12/2019";

int currentPageValue = 0;  
  
Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimCancelListRequest request = new ClaimCancelListRequest();

SearchData searchData = new SearchData();

SearchDate searchDate = new SearchDate();

DateRange dateRange = new DateRange();

RequestPagingData requestPagingData = new RequestPagingData();

requestPagingData.setCurrentPage(currentPageValue);

dateRange.setStartDate(startDate);

dateRange.setEndDate(endDate);

searchDate.setSearchDateType(searchDateType);

searchDate.setPeriod(dateRange);

searchData.setStatus(status);

searchData.setExecuter(executer);

searchData.setSearchInfoType(searchInfoType);

searchData.setSearchQuery(searchQuery);

searchData.setSearchDate(searchDate);

request.setAuthentication(authentication);  
request.setSearchDate (searchData);   
  
ClaimCancelServicePort port = new ClaimCancelServicePortService()  
.getClaimCancelServicePortSoap11();  
ClaimCancelListResponse response = port.claimCancelList(request);

}

### Rejecting the Cancellation Request (ClaimCancelDeny)

Service to reject the cancellation request.

**ClaimCancelDenyRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***claimCancelId*** | ***Id of the cancellation request to be rejected.*** |
| ***denyReasonId*** | ***Id of the reason for rejection. It will be retrieved from*** [***ClaimCancelDenyReasonType***](#_İptal_Ret_Sebepleri) ***service.*** |
| ***denyReasonNote*** | ***Rejection description.*** |

**\*Bold fields are mandatory fields.**

**ClaimCancelDenyResponse**

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

**ClaimCancelDeny Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch: ClaimCancelDenyRequest>

<auth>

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

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

</auth>

<claimCancelId>1234567</claimCancelId>

<denyReasonId>2</denyReasonId>

<denyReasonNote>I rejected</denyReasonNote>

</sch: ClaimCancelDenyRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

</ns3:ClaimCancelDenyResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimCancelDeny Sample Call (JAVA)**

public static void main(String[] args) {

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

Long claimCancelId **= 1234567L;**Long denyReasonId = **2L**;

String denyReasonNote =**"denied"**;

Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimCancelDenyRequest request = new ClaimCancelDenyRequest();

request.setAuthentication(authentication);

request.setClaimCancelId(claimCancelId);

request.setDenyReasonId(denyReasonId);

request.setDenyReasonNote(denyReasonNote);  
  
ClaimCancelServicePort port = new ClaimCancelServicePortService()  
.getClaimCancelServicePortSoap11();  
ClaimCancelDenyResponse response = port.claimCancelList(request);

}

### Listing the Reasons for Cancellation Request Rejection (ClaimCancelDenyReasonType)

Service that lists the reasons for rejection. It will be used in the service to reject the cancellation request ([ClaimCancelDenyRequest](#_İptal_Talebi_Reddetme)).

**ClaimCancelDenyReasonTypeRequest**

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

**\*Bold fields are mandatory fields.**

**ClaimCancelDenyReasonTypeResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *denyReasonTypeDataList.id* | *Id of reason for rejection.* |
| *denyReasonTypeDataList.value* | *Reason for rejection description.* |

**ClaimCancelDenyReasonType Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch: ClaimCancelDenyReasonTypeRequest>

<auth>

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

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

</auth>

</sch: ClaimCancelDenyReasonTypeRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

<denyReasonTypeDataList>

<id>1</id>

<value>Products Out of Stock</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>2</id>

<value>Special Offer Has Ended</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>3</id>

<value>Product Has Been Delivered to Shipping, Intact</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>4</id>

<value>Product is Damaged</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>5</id>

<value>Product Whose Return Cannot Be Accepted As Per Legislation</value>

</denyReasonTypeDataList>

</ns3:ClaimCancelDenyReasonTypeResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimCancelDenyReasonType Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
  
Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimCancelDenyReasonTypeRequest request = new ClaimCancelDenyReasonTypeRequest();

request.setAuthentication(authentication);   
  
ClaimCancelServicePort port = new ClaimCancelServicePortService()  
.getClaimCancelServicePortSoap11();  
ClaimCancelDenyReasonTypeResponse response = port.claimCancelList(request);

}

### Confirming the Cancellation Request (ClaimCancelApprove)

Service to confirm the cancellation request.

**ClaimCancelApproveRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***claimCancelId*** | ***Id of the cancellation request to be approved.*** |

**\*Bold fields are mandatory fields.**

**ClaimCancelApproveResponse**

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

**ClaimCancelApprove Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch: ClaimCancelApproveRequest>

<auth>

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

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

</auth>

<claimCancelId>1234567</claimCancelId>

</sch: ClaimCancelApproveRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

</ns3: ClaimCancelApproveResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimCancelApprove Sample Call (JAVA)**

public static void main(String[] args) {

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

Long claimCancelId **= 1234567L;**  
Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimCancelDenyRequest request = new ClaimCancelDenyRequest();

request.setAuthentication(authentication);

request.setClaimCancelId(claimCancelId);   
  
ClaimCancelServicePort port = new ClaimCancelServicePortService()  
.getClaimCancelServicePortSoap11();  
ClaimCancelApproveResponse response = port.claimCancelList(request);}

## 3.14 Order Return Request Service (ReturnService)

WSDL address for Order Cancellation Requests service:

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

### Return Request Listing (ClaimReturnList)

The service that sends the order return requests to the store as a list. 20 cancellation information is displayed on each page.

**ClaimReturnListRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| *searchData.status* | [*See: ClaimReturnStatus*](#_Sipariş_İade_Talebi) |
| *searchData.executer* | [*See: ClaimExecuterType*](#_İşlemi_Yapan_(Claim) |
| *searchData.searchInfoType* | [*See: ClaimSearchType*](#_İptal_Talebi_Detayı) |
| *searchData.searchQuery* | *The word / code to search by searchData.searchInfoType.* |
| *searchData.period.startDate* | *Return request start date.(dd/mm/yyyy)* |
| *searchData.period.endDate* | *Return request end date.* *(dd/mm/yyyy)* |
| *currentPage* | *The page number shown (currentpage starts with 0).* |

**\*Bold fields are mandatory fields.**

**ClaimReturnListResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *claimReturnList.claimReturn.claimReturnId* | *Return request id.* |
| *claimReturnList.claimReturn.status* | *Return status.* |
| *claimReturnList.claimReturn.executer* | *The information about the person making the transaction* |
| *claimReturnList.claimReturn.returnReasonType* | *Reason for Return* |
| *claimReturnList.claimReturn.returnReasonDescription* | *Return description.* |
| *claimReturnList.claimReturn.orderNumber* | *Order code.* |
| *claimReturnList.claimReturn.sellerCampaignNumber* | *Store Shipment Special Offer No.* |
| *claimReturnList claimReturn.shipmentMethod* | *Shipment Method.* |
| *claimReturnList.claimReturn.bundleName* | *Special Offer Name (If any)* |
| *claimReturnList.claimReturn.requestDate* | *Date of request.* |
| *claimReturnList.claimReturn.pendingRequestDate* | *Postponement Request Date* |
| *claimReturnList.claimReturn.pendedDate* | *Date of Postponement* |
| *claimReturnList.claimReturn.cancelledDate* | *Date of Cancellation.* |
| *claimReturnList.claimReturn.deniedDate* | *Date of Rejection.* |
| *claimReturnList.claimReturn.approvedDate* | *Date of Approval.* |
| *claimReturnList.claimReturn.sellerShipmentCompany* | *Store Shipment Shipping Company* |
| *claimReturnList.claimReturn.sellerTrackingNumber* | *Store Shipment Tracking No.* |
| *claimReturnList.claimReturn.productId* | *Product Code.* |
| *claimReturnList.claimReturn.productName* | *Product Name.* |
| *claimReturnList.claimReturn.attributesNames* | *Product Properties.* |
| *claimReturnList.claimReturn.quantity* | *Number of products with return requests.* |
| *claimReturnList.claimReturn.unitPrice* | *Unit Price.* |
| *claimReturnList.claimReturn.finalPrice* | *Amount Paid.* |
| *claimReturnList.claimReturn.shipmentCompany* | *Shipping Company.* |
| *claimReturnList.claimReturn.trackingNumber* | *Shipment Tracking No.* |
| *claimReturnList.claimReturn.campaignNumber* | *Special Offer No.* |
| *claimReturnList.claimReturn.deliveryFeeType* | *Information about who paid the Shipping Fee;*  *'ByBuyer' Buyer Pays*  *'BySeller' Store Pays*  *‘Conditional’ Conditional Shipment* |
| *claimReturnList.claimReturn.buyerPhone* | *Customer’s Phone Number* |
| *claimReturnList.claimReturn.buyerEmail* | *Customer’s E-mail Address* |
| *claimReturnList.claimReturn.buyerName* | *Customer’s Name & Surname* |
| *claimReturnList.claimReturn.paymentDate* | *Payment Date* |
| *pagingData* | [*See: Pagination*](#_Sayfalandırma_(Pagination)_2) |

**ClaimReturnList Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:ClaimReturnListRequest>

<auth>

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

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

</auth>

<searchData>

<status>ALL</status>

<executer></executer>

<searchInfoType>ORDERID</searchInfoType>

<searchQuery>123456789</searchQuery>

<period>

<startDate></startDate>

<endDate></endDate>

</period>

</searchData>

<pagingData>

<currentPage></currentPage>

</pagingData>

</sch:ClaimReturnListRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

<claimReturnList>

<claimReturn>

<attributesNames/>

<buyerEmail>\*\*\*\*@gmail.com</buyerEmail>

<buyerName>name surname</buyerName>

<buyerPhone>55\*\*\*\*\*</buyerPhone>

<claimReturnId>12345</claimReturnId>

<deliveryFeeType>BySeller</deliveryFeeType>

<finalPrice>100</finalPrice>

<orderNumber>123456789</orderNumber>

<paymentDate>12/09/2019</paymentDate>

<productId>1234</productId>

<productName>Sürat Pop Products</productName>

<quantity>1</quantity>

<requestDate>16/09/2019</requestDate>

<returnReasonDescription>trial</returnReasonDescription>

<returnReasonType>Received a different product</returnReasonType>

<sellerTrackingNumber/>

<shipmentCompany>Sürat</shipmentCompany>

<shipmentMethod>Shipping</shipmentMethod>

<status>REQUESTED</status>

<trackingNumber>4134</trackingNumber>

<unitPrice>100</unitPrice>

</claimReturn>

<pagingData>

<currentPage>0</currentPage>

<pageSize>20</pageSize>

<totalCount>1</totalCount>

<pageCount>1</pageCount>

</pagingData>

</ns3:ClaimReturnListResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimReturnList Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
String status =**"ALL"**;

String searchInfoType =**"ORDERID"**;

String searchQuery =**"123456789"**;

int currentPageValue = 0;  
  
Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimReturnListRequest request = new ClaimReturnListRequest();

SearchData searchData = new SearchData();

DateRange dateRange = new DateRange();

RequestPagingData requestPagingData = new RequestPagingData();

requestPagingData.setCurrentPage(currentPageValue);

dateRange.setStartDate(startDate);

dateRange.setEndDate(endDate);

searchData.setStatus(status);

searchData.setSearchInfoType(searchInfoType);

searchData.setSearchQuery(searchQuery);

searchData.setPeriod(dateRange);

request.setAuthentication(authentication);  
request.setSearchDate (searchData);   
  
ReturnServicePort port = new ReturnServicePortService()  
.getReturnServicePortSoap11();  
ClaimReturnListResponse response = port.claimReturnList(request);

}

### Listing the Reasons for Rejection Request Rejection (ClaimReturnDenyReasonType)

Service that lists the reasons for rejection. It will be used in the service to reject the return request (ClaimReturnDenyRequest).

**ClaimReturnDenyReasonTypeRequest**

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

**\*Bold fields are mandatory fields.**

**ClaimReturnDenyReasonTypeResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *denyReasonTypeDataList.id* | *Id of reason for rejection.* |
| *denyReasonTypeDataList.value* | *Reason for rejection description.* |

**ClaimReturnDenyReasonType Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch: ClaimReturnDenyReasonTypesRequest>

<auth>

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

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

</auth>

</sch: ClaimReturnDenyReasonTypesRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<denyReasonTypeDataList>

<id>21</id>

<value>Special Promotion Product</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>1</id>

<value>Product has been delivered to shipping, intact</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>2</id>

<value>Product is damaged</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>22</id>

<value>Product whose cancellation and return cannot be accepted</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>23</id>

<value>Last 24 hours remaining for the round</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>3</id>

<value>Product Whose Return Cannot Be Accepted As Per Legislation</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>24</id>

<value>Last 48 hours remaining for the round</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>25</id>

<value>Last 72 hours remaining for the round</value>

</denyReasonTypeDataList>

<denyReasonTypeDataList>

<id>41</id>

<value>Return Product Did Not Arrive at the Store</value>

</denyReasonTypeDataList>

</ns3:ClaimReturnDenyReasonTypesResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimReturnDenyReasonType Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
  
Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimReturnDenyReasonTypeRequest request = new ClaimReturnDenyReasonTypeRequest();

request.setAuthentication(authentication);   
  
ReturnServicePort port = new ReturnServicePortService()  
.getReturnServicePortSoap11();  
ClaimReturnDenyReasonTypeResponse response = port.claimReturnList(request);

}

### Listing the Reasons for Postponement Request Rejection (ClaimReturnPendingReasonType)

Service that lists the reasons for postponement. Will be used in the return postponement service ([ClaimReturnPending](#_İade_Talebi_Erteleme_1)).

**ClaimReturnPendingReasonTypeRequest**

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

**\*Bold fields are mandatory fields.**

**ClaimReturnPendingReasonTypeResponse**

|  |  |
| --- | --- |
| *result* | [*See: Result*](#_Sonuç_Durum_Bilgisi(Result)) |
| *pendingReasonTypeDataList.id* | *Postponement reason id.* |
| *pendingReasonTypeDataList.value* | *Reason for postponement description.* |

**ClaimReturnPendingReasonType Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch: ClaimReturnPendingReasonTypesRequest>

<auth>

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

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

</auth>

</sch: ClaimReturnPendingReasonTypesRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<pendingReasonTypeDataList>

<id>1</id>

<value>The product will be inspected by the service</value>

</pendingReasonTypeDataList>

<pendingReasonTypeDataList>

<id>21</id>

<value>The product will be inspected by the agency</value>

</pendingReasonTypeDataList>

<pendingReasonTypeDataList>

<id>2</id>

<value>The product will be inspected by the factory</value>

</pendingReasonTypeDataList>

<pendingReasonTypeDataList>

<id>3</id>

<value>The required checks will be performed</value>

</pendingReasonTypeDataList>

</ns3:ClaimReturnPendingReasonTypesResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimReturnPendingReasonType Sample Call (JAVA)**

public static void main(String[] args) {

String strAppKey = **"\*\*\*"**;  
String strAppSecret = **"\*\*\*"**;  
  
Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimReturnPendingReasonTypeRequest request = new ClaimReturnPendingReasonTypeRequest();

request.setAuthentication(authentication);   
  
ReturnServicePort port = new ReturnServicePortService()  
.getReturnServicePortSoap11();  
ClaimReturnPendingReasonTypeResponse response = port.claimReturnList(request);

}

### Return Request Approval (ClaimReturnApprove)

Service to confirm the return request.

**ClaimReturnApproveRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***claimReturnId*** | ***Id of the return request to be approved.*** |

**\*Bold fields are mandatory fields.**

**ClaimReturnApproveResponse**

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

**ClaimReturnApprove Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch: ClaimReturnApproveRequest >

<auth>

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

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

</auth>

<claimCancelId>1234567</claimCancelId>

</sch: ClaimReturnApproveRequest >

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

</ns3: ClaimReturnApproveResponse >

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimReturnApprove Sample Call (JAVA)**

public static void main(String[] args) {

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

Long claimReturnId **= 1234567L;**  
Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimReturnApproveRequest request = new ClaimReturnApproveRequest();

request.setAuthentication(authentication);

request.setClaimReturnId(claimReturnId);   
  
ReturnServicePort port = new ReturnServicePortService()  
.getReturnServicePortSoap11();  
ClaimReturnApproveResponse response = port.claimReturnList(request);}

### Return Request Postponement (ClaimReturnPending)

Return request postponement service.

**ClaimReturnPendingRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***claimReturnlId*** | ***Id of the return request to be postponed.*** |
| ***pendingReasonId*** | ***Id of the reason for postponement. It will be retrieved from*** [***ClaimReturnPendingReasonType***](#_İade_Talebi_Erteleme) ***service.*** |
| ***pendingDayCount*** | ***Number of Postponement Days.*** |
| ***pendingReasonNote*** | ***Postponement Description*** |

**\*Bold fields are mandatory fields.**

**ClaimReturnPendingResponse**

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

**ClaimReturnPending Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:ClaimReturnPendingRequest>

<auth>

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

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

</auth>

<claimReturnId>123456789</claimReturnId>

<pendingReasonId>123</pendingReasonId>

<pendingDayCount>1</pendingDayCount>

<pendingReasonNote>description</pendingReasonNote>

</sch:ClaimReturnPendingRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

</ns3: ClaimReturnPendingResponse >

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimReturnPending Sample Call (JAVA)**

public static void main(String[] args) {

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

Long claimReturnId **= 123456789L;**

Long pendingReasonId**= 123L;**

Integer pendingDayCount **= 1;**

String pendingReasonNote **= "description";**

Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimReturnPendingRequest request = new ClaimReturnPendingRequest();

request.setAuthentication(authentication);

request.setClaimReturnId(claimReturnId);

request.setPendingReasonId(pendingReasonId);

request.setPendingDayCount(pendingDayCount);

request.setPendingReasonNote(pendingReasonNote);  
  
ReturnServicePort port = new ReturnServicePortService()  
.getReturnServicePortSoap11();  
ClaimReturnPendingResponse response = port.claimReturnList(request);}

### Return Request Rejection (ClaimReturnDeny)

Service to reject the return request.

**ClaimReturnDenyRequest**

|  |  |
| --- | --- |
| ***Auth*** | [*See: Authentication*](#_2.1_Güvenlik_Mekanizması) |
| ***claimReturnlId*** | ***Id of the return request to be rejected.*** |
| ***denyReasonId*** | ***Id of the reason for rejection It will be retrieved from*** [***ClaimReturnDenyReasonType***](#_İade_Talebi_Ret) ***service.*** |
| ***denyReasonNote*** | ***Rejection Description.*** |
| *returnShipmentType* | *Shipment Type.*  *"CUSTOM" = With Tracking Number*  *"CAMPAIGN" = With Shipping Special Offer*   * *In the rejection of return requests for Same-Day Delivery, Special Delivery and Ticket orders;* ***Leave this field blank.*** * *If the selected rejection reason is “****Product Did Not Arrive at the Store****”;* ***Leave this field empty.*** *The tracking number entered by the receiver will be sent automatically.* * *For other rejection reasons;* * *If “CUSTOM” is selected; if you have delivered the product to the shipping company, please enter the tracking number into the corresponding field.* * *If “CAMPAIGN” is selected;* ***please leave the following fields empty: shipmentCompanyId , trackingNumber , deliveryReceiptNumber.*** *If you have not shipped the product for return yet, when you send the product along with the special offer code and by the specified shipping company, the tracking code is automatically identified in the system and your return request is automatically approved. If you approve, you can access the shipment special offer codes that will be generated for the products from the “Store Shipment Special Offer No” column of the table.* |
| *shipmentCompanyId* | *Shipping company Id.* It will be retrieved from [*ShipmentCompanyService*](#_3.7_Kargo_Şirketi) *service.*  *If the returnShipmentType is “****CUSTOM****” and reason for Rejection is not “****Product Did Not Arrive at the Store****”;* ***This field is mandatory.***  *For other cases;* ***Leave this field blank.*** |
| *trackingNumber* | *Tracking Number.*  *If the returnShipmentType is “****CUSTOM****” and reason for Rejection is not “****Product Did Not Arrive at the Store****”;* ***This field is mandatory.***  *For other cases;* ***Leave this field blank.*** |
| *deliveryReceiptNumber* | *Delivery Receipt No.*  *In the rejection of return requests for Same-Day Delivery, Special Delivery and Ticket orders;* ***This field is mandatory.***  *For other cases;* ***Leave this field blank.*** |

**\*Bold fields are mandatory fields.**

**ClaimReturnDenyResponse**

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

**ClaimReturnDeny Sample Call**

**Request**

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

<soapenv:Header/>

<soapenv:Body>

<sch:ClaimReturnDenyRequest>

<auth>

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

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

</auth>

<claimReturnId>123456789</claimReturnId>

<denyReasonId>123</denyReasonId>

<denyReasonNote>same-day delivery return request rejection</denyReasonNote>

<returnShipmentType></returnShipmentType>

<shipmentCompanyId></shipmentCompanyId>

<trackingNumber></trackingNumber>

<deliveryReceiptNumber>123456</deliveryReceiptNumber>

</sch:ClaimReturnDenyRequest>

</soapenv:Body>

</soapenv:Envelope>

**Response**

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

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

<result>

<status>success</status>

</result>

</ns3: ClaimReturnDenyResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**ClaimReturnDeny Sample Call (JAVA)**

public static void main(String[] args) {

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

Long claimReturnId **= 123456789L;**

Long denyReasonId**= 123L;**

String denyReasonNote **= "same-day delivery return request rejection";**

String deliveryReceiptNumber **= "123456"**

Authentication authentication = new Authentication();  
authentication.setAppKey(strAppKey);  
authentication.setAppSecret(strAppSecret);  
  
ClaimReturnPendingRequest request = new ClaimReturnPendingRequest();

request.setAuthentication(authentication);

request.setClaimReturnId(claimReturnId);

request.setDenyReasonId(denyReasonId);

request.setDenyReasonNote(denyReasonNote);

request.setDeliveryReceiptNumber(deliveryReceiptNumber);  
  
ReturnServicePort port = new ReturnServicePortService()  
.getReturnServicePortSoap11();  
ClaimReturnDenyResponse response = port.claimReturnList(request);}

## 3.15 Commonly Used Data Templates

### Authorization (Authentication)

|  |  |
| --- | --- |
| *auth.appKey* | *Application key to use for accessing services* |
| *auth.appSecret* | *Application password to use for accessing services* |

### Result Status Information (Result)

|  |  |
| --- | --- |
| *result.status* | *Shows transaction result status information. Retrieves "success" and "failure" status information.* |

### Pagination

|  |  |
| --- | --- |
| *pagingData.currentPage* | *Page number shown (currentpage starts with 0)* |
| *pagingData.pageSize* | *Number of items on the page* |
| *pagingData.totalCount\** | *Total number of items* |
| *pagingData.pageCount\** | *Total number of pages* |

\* It is not used when making a request, it returns to the calling party in response.

### Product Summary Information (Product Summary Data)

|  |  |
| --- | --- |
| *product.currencyAmount* | *Currency amount of the displayed product price* |
| *product.currencyType* | *Currency of the displayed product price*  *1: TRY*  *2: USD*  *3: EUR* |
| *product.displayPrice* | *Displayed product price (The final price of the product as a result of discounts)* |
| *product.id* | *Product N11 ID* |
| *product.productSellerCode* | *Store product code* |
| *product.title* | *Product title* |
| *product.subtitle* | *Product subtitle* |
| *product.oldPrice* | *Old base price of the product (Returns after updated methods)* |
| *product.price* | *Base price of the product* |
| *product.displayPrice* | *Displayed product price (The final price of the product as a result of discounts)* |
| *product.saleStatus* | *1: Before Sales(Before\_Sale)*  *2: On sale(On\_Sale)*  *3: Out of Stock (Out of\_Stock)*  *4: Closed for sale (Sale\_Closed)* |
| *product.approvalStatus* | *Product approval status:*  *1: Active (On sale)*  *2: Pending*  *3: Prohibited* |
| *product.stockItems.stockItem.id* | *Product stock N11 ID* |
| *product.stockItems.stockItem.quantity* | *Product stock quantity* |
| *product.stockItems.stockItem.version* | *Product stock version number* |
| *product.stockItems.stockItem.currencyAmount* | *Amount of product stock unit in currency* |
| *product.stockItems.stockItem.displayPrice* | *The displayed product price of the product stock unit (the final price defined as a result of the seller discount of the product)* |
| product.stockItems.stockItem.bundle | *Products that are sold together* |
| product.stockItems.stockItem.mpn | *Manufacturer part number of the product* |
| product.stockItems.stockItem.gtin | *Global trade item number of the product* |
| product.stockItems.stockItem.oem | *Part/Product connection number* |
| product.stockItems.stockItem.n11CatalogId | *Product’s N11 catalog number* |
| *product.stockItems.stockItem.sellerStockCode* | *Product stock store code* |
| *product.stockItems.stockItem.oldOptionPrice* | *Previous list price of product stock unit (Returns after updated methods)* |
| *product.stockItems.stockItem.optionPrice* | *List price of the product stock unit* |
| *product.stockItems.stockItem.attributes.attribute.name* | *Option name with stock definition* |
| *product.stockItems.stockItem.attributes.attribute.value* | *Option value with stock definition* |

### Product Sale Status (Product Sale Status)

|  |  |
| --- | --- |
| *product.saleStatus* | *1: Before Sales (Before\_Sale)*  *2: On Sale (On\_Sale)*  *3: Out of Stock (Out of\_Stock)*  *4: Closed for sale (Sale\_Closed)* |

### Product Approval Status (Product Approval Status)

|  |  |
| --- | --- |
| *product.approvalStatus* | *Product approval status:*  *1: Active (On sale)*  *2: Pending*  *3: Prohibited*  *4: Unlisted*  *5: Pending approval*  *6: Rejected* |

### Product Discount Information (Product Discount Data)

|  |  |
| --- | --- |
| *discount.discountType* | Store Discount Type:  1: As Discount Amount  2: As Discount Rate  *3: As Discounted Price* |
| *discount.discountValue* | *Discount Amount (according to the parameter given in the discount type)* |
| *discount.discountStartDate* | *Store discount start date* |
| *discount.discountEndDate* | *Store discount end date* |

### Product Status (Product Condition)

|  |  |
| --- | --- |
| *product.productCondition* | *Product status:*  *1: New*  *2: 2nd hand* |

### Approval Status (Approval Status)

|  |  |
| --- | --- |
| productSearch.approvalStatus | Product approval status  “Active”: Products on sale  “Suspended”: Pending products  “Prohibited”: Prohibited products  “Unlisted”: Unlisted products  “WaitingForApproval”: Products pending approval  “Rejected”: Rejected products  “UnapprovedUpdate”: Products awaiting update confirmation |

### Product Title (Title)

|  |  |
| --- | --- |
| **product.title** | The title that will appear on the product page must be 65 characters or less. |

### Product Subtitle (Subtitle)

|  |  |
| --- | --- |
| **product.subtitle** | The subtitle that will appear on the product page must be 65 characters or less. |

### Discount Type (Discount Type)

|  |  |
| --- | --- |
| ***productDiscount.discountType*** | Store Discount Type:  1: As Discount Amount  2: As Discount Rate  *3: As Discounted Price* |

### Order Status (SearchData)

|  |  |
| --- | --- |
| *searchData.status* | *Order status:*  *“New”: New orders*  *“Approved”: Approved Orders*  *“Rejected”: Rejected Orders*  *“Shipped”: Shipped Orders*  *“Delivered”: Delivered Orders*  *“Completed”: Completed Orders*  *“Claimed”: Orders in Cancellation/Return/Exchange Status*  "LATE\_SHIPMENT" Orders whose Shipping Delayed |

### Payment Type (Payment Type)

|  |  |
| --- | --- |
| *orderList.order.paymentType* | *Payment type:*  *1: Credit Card*  *2: BKMEXPRESS*  *3: AKBANKDIREKT*  *4: PAYPAL*  *5: MallPoint*  *6: GARANTIPAY*  *7: GarantiLoan*  *8: MasterPass*  *9: ISBANKPAY*  *10: PAYCELL*  *11: COMPAY*  *12: YKBPAY*  *13: FIBABANK*  *14: Other* |

### Shipment Payment Method (Delivery Fee Type)

|  |  |
| --- | --- |
| *orderList.order.orderItemList.orderItem.deliveryFeeType* | *Shipment payment method:*  *1: N11 Pays*  *2: Buyer Pays*  *3: Store Pays*  *4: Conditional Shipment (Buyer Pays)*  *5: Conditional Shipping Free (Seller Pays)* |

### Service Order Item Type (Orderitem Type)

|  |  |
| --- | --- |
| *orderDetail.serviceItemList.serviceItem.orderItemType* | *Order Item Type:*  *1: Product Order Item*  *2: Service Order Item*  *3: Gsm Order Item*  *4: Travel Order Item* |

### Order Status Information (Order Status)

|  |  |
| --- | --- |
| *orderList.order.status* | Order status information:  1:Pending Transaction  2:Processing  3:Rejected  4:Invalid  5:Completed |

### Order Item Status Information (OrderItem Status)

|  |  |
| --- | --- |
| *orderList.order.orderItemList.orderItem.status* | Order Item Status Information:  1:Pending Transaction  2:Paid  3:Invalid  4:Rejected  5:Accepted  6:In shipment  *7:Delivered*  *8:Rejected*  *9:Returned*  *10:Completed*  *11: Return Cancellation Change Requested*  *12: Return Cancellation Change Completed*  *13:In Shipment Return*  *14:Delayed Shipping,*  *15: Accepted But Not Shipped On Time*  *16: Delivered Return,*  *17:Return After Completion* |

### Payment Type (Payment Type)

|  |  |
| --- | --- |
| *orderDetail.paymentType* | *Payment type:*  *1: Credit Card*  *2: BKMEXPRESS*  *3: AKBANKDIREKT*  *4: PAYPAL*  *5: MallPoint*  *6: GARANTIPAY*  *7: GarantiLoan*  *8: MasterPass*  *9: ISBANKPAY*  *10: PAYCELL*  *11: COMPAY*  *12: YKBPAY*  *13: Other* |

**SortForUpdate**

|  |  |
| --- | --- |
| *searchData.sortForUpdateDate* | *1.true (Lists orders previously created but updated to the specified date)*  *2.false (Lists orders created on the specified date)* |

### Shipment Multiple Shipment Status (Shipment Allowed)

|  |  |
| --- | --- |
| *shipmentApiModel.combinedShipmentAllowed* | *If the shipping status of more than 1 product is true, it means that there is more than 1 product; If false, it is the only product* |

### Order Cancellation Request Status (Claim Cancel Status)

|  |  |
| --- | --- |
| searchData.status | Order cancellation request status. If it is sent blank or a value other than the following types are sent, orders with the status "REQUESTED" are displayed.  “REQUESTED”: Cancellation Requested  “RETRACTED”: Cancellation Withdrawn  “COMPLETED”: Cancelled  “DENIED”: Cancellation Request Rejected  “REJECT”: Rejected  “MANUAL\_REFUND” : Manuel Refund Completed  “ALL”: All cancellation statuses |

### 

### Transaction Executer (Claim Executer Type)

|  |  |
| --- | --- |
| searchData.executer | The information about the person making the transaction If a value other than the following types is sent, it is considered blank.  “BUYER”: Buyer  “SELLER”: Seller  “OPERATOR”: Operator  “SYSTEM”: System  “SHIPMENT”: Shipping Company |

### Cancellation Request Detail (Claim Search Type)

|  |  |
| --- | --- |
| searchData.searchInfoType | Cancellation request details should be sent if search is to be made with searchData.searchQuery. Specifies the type of search detail. If a value other than the following types is sent, it is considered blank.  “BUYEREMAIL”: Buyer’s E-mail Address  “RECIPIENTS”: Recipient  “BUYERNAME”: Buyer’s Name & Surname  “ORDERID”: Order Code  “PRODUCTID”: Product Code |

### Cancellation Request Search Date Type (Claim Cancel Date Type)

|  |  |
| --- | --- |
| searchData.searchDate.searchDateType | The cancellation request specifies the type of searchData.searchDate.period field entered in the search.  If it is sent blank or a value other than the following types is sent, it is considered as "REQUESTED".  “REQUESTED”: Date of request  “ACCEPTED”: Acceptance Date  “DENIED”: Rejection Date |

### Order Return Request Status (Claim Return Status)

|  |  |
| --- | --- |
| searchData.status | Order return request status. If it is sent blank or a value other than the following types are sent, orders with the status "REQUESTED" are displayed.  “REQUESTED”: Return Requested  “CANCELLED”: Cancelled  “DENIED”: Rejected  “PENDING”: Postponement Requested  “PENDED”: Evaluation Postponed  “APPROVED”: Return Approved  “MANUAL\_REFUND” : Manuel Refund Completed  “ALL”: All return statuses |

prepared by Dogus Planet