

Introduction

The Store Web Service gives merchants access to 3dCart Stores information. With the Store Web Service APIs, you can build applications to perform tasks such as listing or update orders, customers, and products.

Table of Contents

TABLE OF CONTENTS.....	2
GETTING STARTED	3
WEB SERVICES METHODS	4
1. CUSTOMER RELATED METHODS	4
1.1. GETCUSTOMER	4
1.2. GETCUSTOMERCOUNT.....	5
1.3. EDITCUSTOMER.....	6
2. ORDER RELATED METHODS	9
2.1. GETORDER	9
2.2. GETORDERCOUNT.....	11
2.3. GETORDERSTATUS.....	12
2.4. UPDATEORDERSTATUS	13
2.5. UPDATEORDERSHIPMENT.....	14
3. PRODUCT RELATED METHODS	16
3.1. GETPRODUCT.....	16
3.2. GETPRODUCTCOUNT	19
3.3. GETPRODUCTINVENTORY	20
3.4. UPDATEPRODUCTINVENTORY	20
WORKING WITH THE CALLBACKURL PARAMETER	22
WEB SERVICES ERRORS.....	23
END - YOU'VE REACHED THE END OF THIS GUIDE.	25
CONTACTING 3DCART	25

Getting Started

In order to use the Store Web Services, you first need to configure the API Settings in the Online Store Manager (Settings -> General -> API Settings).

Settings:

- Enable API: Enable/Disable the use of the API.
- API User Key: Auto-generated 32 character string required to authenticate the APIs requests.
- Admin User: Identity used by the web services to make the requests.
- IP Address: You can restrict access to the API to one or more IP. If none is entered, the API will accept requests from any IP.

The Store Web Services can be accessed in the following URL: <http://api.3dcart.com/cart.asmx>

Web Services Methods

1. Customer Related Methods

1.1. getCustomer

Description: Get customer information.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
batchSize *	Numeric	Number of records to pull. Range: 1 to 100.
startNum *	Numeric	Position to start the search. Range: 1 to x
customersFilter	String	Comma delimited string with zero or more search parameters. Allowed parameters: firstname, lastname, email, countrycode, statecode, city, phone. i.e.: firstname=John,email=john@email.com, countrycode=US,statecode=FL,city=Margate
callBackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getCustomer xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <batchSize>1</batchSize>
      <startNum>1</startNum>
      <customersFilter>firstname=John</customersFilter>
      <callBackURL></callBackURL>
    </getCustomer>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<CustomersRequestResponse xmlns="">
  <Customer>
    <CustomerID>29</CustomerID>
    <UserID />
    <Password>123456</Password>
  </Customer>
  <BillingAddress>
    <FirstName>John</FirstName>
    <LastName>Brown</LastName>
```

```
<Address>111 BRANCH MORR DR</Address>
<Address2 />
<City>Margate</City>
<StateCode>FL</StateCode>
<ZipCode>33063</ZipCode>
<CountryCode>US</CountryCode>
<Company />
<Phone>1234567897</Phone>
<Email>john@email.com</Email>
</BillingAddress>
<ShippingAddress>
  <FirstName>John</FirstName>
  <LastName>Brown</LastName>
  <Address>111 BRANCH MORR DR </Address>
  <Address2 />
  <City>Margate</City>
  <StateCode>FL</StateCode>
  <ZipCode>33063</ZipCode>
  <CountryCode>US</CountryCode>
  <Company />
  <Phone>12345678945</Phone>
</ShippingAddress>
<Comments />
<LastLoginDate>2/17/2007</LastLoginDate>
<WebSite />
<DiscountGroup />
<CustOther1 />
<AccountNumber>1333</AccountNumber>
<MailList>1</MailList>
<CustomerType>0</CustomerType>
<LastUpdate>2/17/2007</LastUpdate>
<CustEnabled>1</CustEnabled>
<AdditionalFields>
  <AdditionalField1 />
  <AdditionalField2 />
  <AdditionalField3 />
  <AdditionalField4 />
</AdditionalFields>
</Customer>
</CustomersRequestResponse>
```

1.2. getCustomerCount

Description: Get total customers.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
callBackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getCustomerCount xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <callbackURL></callbackURL>
    </getCustomerCount>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<CustomerCountResponse xmlns="">
  <CustomerCount>15473</CustomerCount>
</CustomerCountResponse>
```

1.3. editCustomer

Description: Edit a customer record.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
customerData*	String	Name-value pair containing all customer's information. This string must be in the following format: name_of_the_field1===value_of_the_field1 name_of_the_field2===value_of_the_field2 name_of_the_field3===value_of_the_field3... Field and value separated by: === Each pair separated by: See customerData specification below for a complete reference of the fields.
action*	String	Indicates if the customer should be inserted, updated or deleted. This string must be one of the following: insert, update or delete. If action is delete, just contactid or alt_contactid is required on the customerData field.
callbackURL	String	URL where the XML response will be posted.

* Required parameters

customerData specification:

Name	Type (length)	Description
contactid	Numeric	Customer id - identifies the customer. Required when alt_contactid is blank.
alt_contactid	String (50)	Alternative customer id - identifies the customer. If has value, contactid will be ignored.
billing_firstname	String (50)	Billing first name. Required when action = insert.
billing_lastname	String (50)	Billing last name. Required when action = insert.
billing_address	String (255)	Billing address 1.
billing_address2	String (50)	Billing address 2.
billing_city	String (100)	Billing city.
billing_state	String (100)	Billing state (2 digits for US states).
billing_zip	String (20)	Billing zip code.
billing_country	String (2)	Billing country (ISO Country Code).
billing_company	String (255)	Billing company.
billing_phone	String (50)	Billing phone.
email	String (100)	Email. Required when action = insert.
shipping_firstname	String (50)	Shipping first name.
shipping_lastname	String (50)	Shipping last name.
shipping_address	String (255)	Shipping address 1.
shipping_address2	String (50)	Shipping address 2.
shipping_city	String (100)	Shipping city.
shipping_state	String (100)	Shipping state (2 digits for US states).
shipping_zip	String (20)	Shipping zip code.
shipping_country	String (2)	Shipping country (ISO Country Code).
shipping_company	String (255)	Shipping company.
shipping_phone	String (50)	Shipping phone.
comments	String (255)	Comments.
website	String (200)	Website.
pass	String (20)	Password. Required when action = insert.
discount	Numeric	Customer group id (0 for none). Default = 0
accountno	String (50)	Tax id.
maillist	Numeric	Indicates if the customer is subscribed on the mailing list. Must be: 0 or 1. Default = 0
custenabled	Numeric	Indicates if the customer is enabled. Must be: 0 or 1. Default = 0
additional_field1	String (250)	Additional field 1.

additional_field2	String (250)	Additional field 2.
additional_field3	String (150)	Additional field 3.

*Please note: every field passed on customerData will be updated.

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <editCustomer xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <customerData>contactid===123|||billing\_firstname===John</customerData>
      <action>update</action>
      <callBackURL></callBackURL>
    </editCustomer>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<EditCustomerResponse xmlns="">
  <result>OK</ result >
  <contactid>123</contactid>
  <alt_contactid></alt_contactid>
</EditCustomerResponse>
```


2. Order Related Methods

2.1. getOrder

Description: Get order information.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
batchSize *	Numeric	Number of records to pull. Range: 1 to 100.
startNum *	Numeric	Position to start the search. Range: 1 to x
startFrom *	Boolean	If startFrom is true and invoiceNum is specified, the web service will return orders >= invoiceNum. If startFrom is false and invoiceNum is specified, the web service will return just the specified order. If invoiceNum is not specified, this parameter will be ignored.
invoiceNum	String	Search for specific invoice number.
status	String	Search orders by status.
dateFrom	Date	Search orders that were placed after specified date. Must be in mm/dd/yyyy format.
dateTo	Date	Search orders that were placed before specified date. Must be in mm/dd/yyyy format.
callBackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getOrder xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <batchSize>1</batchSize>
      <startNum>1</startNum>
      <startFrom>false</startFrom>
      <invoiceNum></invoiceNum>
      <status></status>
      <dateFrom>08/01/2008</dateFrom>
      <dateTo></dateTo>
      <callBackURL></callBackURL>
    </getOrder>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetOrdersResponse xmlns="">
  <Order>
    <OrderID>35446</OrderID>
    <InvoiceNumber>AB-1426</InvoiceNumber>
    <CustomerID>15427</CustomerID>
    <Date>8/4/2008</Date>
    <Total>1590</Total>
    <Tax>90.00</Tax>
    <Tax2>0.00</Tax2>
    <Tax3>0.00</Tax3>
    <Shipping>0</Shipping>
  <BillingAddress>
    <FirstName>Sam</FirstName>
    <LastName>Jones</LastName>
    <Email>sam@email.com</Email>
    <Address>1234 Holmberg Rd</Address>
    <Address2 />
    <City>Parkland</City>
    <ZipCode>33067</ZipCode>
    <StateCode>FL</StateCode>
    <CountryCode>US</CountryCode>
    <Phone>1112223344</Phone>
    <Company />
  </BillingAddress>
  <Comments>
    <OrderComment />
    <OrderInternalComment></OrderInternalComment>
    <OrderExternalComment />
  </Comments>
  <PaymentMethod>Online Credit Card</PaymentMethod>
  <Time>6:12:22 PM</Time>
  <Transaction>
    <CVV2>123</CVV2>
    <ResponseText></ResponseText>
    <AVS>AVS Result</AVS>
    <TransactionId>123456</TransactionId>
    <ApprovalCode>123456</ApprovalCode>
    <TransactionType>Authorize</TransactionType>
    <Amount>1,000.00</Amount>
  </Transaction>
  <Discount>0.00</Discount>
  <Promotions />
  <GiftCertificatePurchased />
  <GiftCertificateUsed />
  <OrderStatus>New</OrderStatus>
  <Referer>http://www.google.com</Referer>
  <SalesPerson />
  <IP>111.222.333.444</IP>
  <DateStarted>8/4/2008 5:54:18 PM</DateStarted>
  <UserID />
  <LastUpdate>8/4/2008</LastUpdate>
  <Weight>0.00</Weight>
  <AffiliateInformation>
    <AffiliateID>0</AffiliateID>
    <AffiliateCommission>0</AffiliateCommission>
    <AffiliateApproved />
    <AffiliateApprovedreason />
  </AffiliateInformation>
  <ShippingInformation>
```

```
<Shipment>
  <ShipmentID>1</ShipmentID>
  <ShipmentDate />
  <Shipping>0</Shipping>
  <Method>Free Shipping</Method>
  <FirstName>Sam</FirstName>
  <LastName>Jones</LastName>
  <Company />
  <Address>1234 Holmberg Rd</Address>
  <Address2 />
  <City>Parkland</City>
  <ZipCode>33067</ZipCode>
  <StateCode>FL</StateCode>
  <CountryCode>US</CountryCode>
  <Phone>1112223344</Phone>
  <Weight>0.00</Weight>
  <Status>New</Status>
  <InternalComment></InternalComment>
  <TrackingCode />
</Shipment>
<OrderItems>
  <Item>
    <ShipmentID>1</ShipmentID>
    <ProductID>100514</ProductID>
    <ProductName>This is the product name – Free Product</ProductName>
    <Quantity>1</Quantity>
    <UnitPrice>0.00</UnitPrice>
    <UnitCost>0.00</UnitCost>
    <OptionPrice>0.00</OptionPrice>
    <Weight>0.00</Weight>
    <WarehouseID>0</WarehouseID>
    <DateAdded />
    <PageAdded />
  </Item>
  <Item>
    <ShipmentID>1</ShipmentID>
    <ProductID>abc3</ProductID>
    <ProductName>Product Name</ProductName>
    <Quantity>1</Quantity>
    <UnitPrice>1,500.00</UnitPrice>
    <UnitCost>7.00</UnitCost>
    <OptionPrice>0.00</OptionPrice>
    <Weight>32.00</Weight>
    <WarehouseID>2</WarehouseID>
    <DateAdded>8/4/2008 5:56:43 PM</DateAdded>
    <PageAdded>Product-Name_p_1234-3214.html</PageAdded>
  </Item>
</OrderItems>
</ShippingInformation>
</Order>
</GetOrdersResponse>
```

2.2. getOrderCount

Description: Get total orders.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
startFrom *	Boolean	If startFrom is true and invoiceNum is specified, the web service will count orders >= invoiceNum. If startFrom is false and invoiceNum is specified, the web service will count just the specified order. If invoiceNum is not specified, this parameter will be ignored.
invoiceNum	String	Search for specific invoice number.
status	String	Search orders by status.
dateFrom	Date	Search orders that were placed after specified date. Must be in mm/dd/yyyy format.
dateTo	Date	Search orders that were placed before specified date. Must be in mm/dd/yyyy format.
callBackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getOrderCount xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <startFrom>false</startFrom>
      <invoiceNum></invoiceNum>
      <status>New</status>
      <dateFrom>08/01/2008</dateFrom>
      <dateTo>08/31/2008</dateTo>
      <callBackURL></callBackURL>
    </getOrderCount>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<OrdersCountResponse xmlns="">
  <Quantity>474</Quantity>
</OrdersCountResponse>
```

2.3. getOrderStatus

Description: Get the status of a specific order.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
invoiceNum *	String	Search for specific invoice number.
callBackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getOrderStatus xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <invoiceNum>1476</invoiceNum>
      <callBackURL></callBackURL>
    </getOrderStatus>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<OrderStatusResponse xmlns="">
  <InvoiceNum>1476</InvoiceNum>
  <StatusId>1</StatusId>
  <StatusText>New</StatusText>
</OrderStatusResponse>
```

2.4. updateOrderStatus

Description: Update the status of a specific order.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
invoiceNum *	String	Invoice number to be updated.
newStatus *	String	New status for the specified order.
callBackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <updateOrderStatus xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <invoiceNum>1476</invoiceNum>
      <newStatus>New</newStatus>
      <callBackURL></callBackURL>
    </updateOrderStatus>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<UpdateOrderStatusResponse xmlns="">
  <InvoiceNum>1476</InvoiceNum>
  <NewStatus>new</NewStatus>
</UpdateOrderStatusResponse>
```

2.5. updateOrderShipment

Description: Update the tracking number and the shipping date of a specific order.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
invoiceNum *	String	Invoice number to be updated.
shipmentID	Numeric	Identifies the shipment id for multiple shipment orders. This ID can be found on the response of the getOrder method. * This field is required for multiple shipment orders.
tracking *	String	Tracking code of the specified order/shipment.
shipmentDate *	Date	Shipping date of the specified order/shipment.
callBackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <updateOrderShipment xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
```

```
<userKey>12345678912345678912345678912345</userKey>
<invoiceNum>1476</invoiceNum>
<shipmentID></shipmentID>
<tracking>1234567890</tracking>
<shipmentDate>01/20/2009</shipmentDate>
<callBackURL></callBackURL>
</updateOrderShipment>
</soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<UpdateOrderShipmentResponse xmlns="">
  <result>OK</result>
</UpdateOrderShipmentResponse>
```

3. Product Related Methods

3.1. getProduct

Description: Get product information.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
batchSize *	Numeric	Number of records to pull. Range: 1 to 100.
startNum *	Numeric	Position to start the search. Range: 1 to x
productId	String	Search for specific product id.
callBackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProduct xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <batchSize>1</batchSize>
      <startNum>1</startNum>
      <productId></productId>
      <callBackURL></callBackURL>
    </getProduct>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetProductDetailsResponse xmlns="">
  <Product>
    <ProductID>prod123</ProductID>
    <ProductName>Test Product</ProductName>
  </Product>
  <Categories>
    <Category>
      <CategoryID>27704</CategoryID>
      <CategoryName>Packaging</CategoryName>
    </Category>
    <Category>
      <CategoryID>25848</CategoryID>
      <CategoryName>Test Sub</CategoryName>
    </Category>
  </Categories>
  <Mfgid>1221</Mfgid>
  <Manufacturer>Manufacturer A</Manufacturer>
  <Distributor>3dcart</Distributor>
```



```
<Cost>154.00</Cost>
<Price>261.00</Price>
<RetailPrice>462</RetailPrice>
<SalePrice>0.00</SalePrice>
<OnSale>0</OnSale>
<Stock>10</Stock>
<StockAlert>0</StockAlert>
<Weight>1.00</Weight>
<Width>0.00</Width>
<Height>0.00</Height>
<Depth>0.00</Depth>
<MinimumOrder>1</MinimumOrder>
<MaximumOrder>0</MaximumOrder>
<DateCreated>9/1/2008</DateCreated>
<Description>Test product description</Description>
<ExtendedDescription>Product description...</ExtendedDescription>
<Keywords>keyword, test, product, buy, store</Keywords>
<RelatedProducts />
<ShipCost>0.00</ShipCost>
<Title>Product title...</Title>
<MetaTags />
<DisplayText />
<HomeSpecial>0</HomeSpecial>
<CategorySpecial>0</CategorySpecial>
<Hide>0</Hide>
<FreeShipping>0</FreeShipping>
<NonTax>0</NonTax>
<NotForsale>0</NotForsale>
<GiftCertificate>0</GiftCertificate>
<UserId>John Admin</UserId>
<LastUpdate>8/1/2008</LastUpdate>
<ExtraFields>
<ExtraField1 />
<ExtraField2 />
<ExtraField3 />
<ExtraField4 />
<ExtraField5 />
<ExtraField6 />
<ExtraField7 />
<ExtraField8 />
<ExtraField9 />
<ExtraField10 />
</ExtraFields>
<UseCatoptions>0</UseCatoptions>
<QuantityOptions />
<PriceLevel>
<Price_1>0</Price_1>
<Price_2>0</Price_2>
<Price_3>0</Price_3>
<Price_4>0</Price_4>
<Price_5>0</Price_5>
<Price_6>0</Price_6>
<Price_7>0</Price_7>
<Price_8>0</Price_8>
<Price_9>0</Price_9>
<Price_10>0</Price_10>
</PriceLevel>
<MinOrder>0</MinOrder>
<ListingDisplayType>-1</ListingDisplayType>
<ShowOutStock>-1</ShowOutStock>
```

```
<PricingGroupOpt>0</PricingGroupOpt>
<QuantityDiscountOpt>0</QuantityDiscountOpt>
<LoginLevel>-1</LoginLevel>
<RedirectTo />
<AccessGroup />
<SelfShip>0</SelfShip>
<TaxCode />
<eProduct>
  <eProductPassword />
  <eProductRandom>0</eProductRandom>
  <eProductExpire>0</eProductExpire>
  <eProductPath />
  <eProductSerial>0</eProductSerial>
  <eProductInstructions />
  <eProductReuseSerial>0</eProductReuseSerial>
</eProduct>
<NonSearchable>0</NonSearchable>
<InstockMessage />
<OutOfStockMessage />
<BackOrderMessage />
<Rewards>
  <RewardPoints>0</RewardPoints>
  <RewardDisable>0</RewardDisable>
  <RewardRedeem>2610</RewardRedeem>
</Rewards>
<FileName></FileName>
<Images>
  <Image>
    <Url>assets/images/prodtest.jpg</Url>
    <Caption />
  </Image>
  <Image>
    <Url />
    <Caption />
  </Image>
  <Image>
    <Url />
    <Caption />
  </Image>
  <Image>
    <Url />
    <Caption />
  </Image>
  <Thumbnail>assets/images/thumb_prodtest.jpg</Thumbnail>
</Images>
<Options>
  <Option>
    <Id>11</Id>
    <OptionType>Size</OptionType>
  <Values>
    <Value>
      <ID>28</ID>
      <Name>Small</Name>
      <OptionPrice>0.00</OptionPrice>
      <OptionPartNumber />
    </Value>
    <Value>
      <ID>29</ID>
      <Name>Large</Name>
      <OptionPrice>0.00</OptionPrice>
```

```
<OptionPartNumber />
</Value>
</Values>
</Option>
<Option>
  <Id>12</Id>
  <OptionType>Color</OptionType>
  <Values>
    <Value>
      <ID>30</ID>
      <Name>Blue</Name>
      <OptionPrice>0.00</OptionPrice>
      <OptionPartNumber />
    </Value>
  </Values>
</Option>
</Options>
</Product>
</GetProductDetailsResponse>
```

3.2. getProductCount

Description: Get total products.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
callbackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductCount xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <callbackURL></callbackURL>
    </getProductCount>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetProductCountResponse xmlns="">
  <ProductQuantity>6041</ProductQuantity>
```

```
</GetProductCountResponse>
```

3.3. getProductInventory

Description: Get the stock of a specific product.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com
userKey *	String	Auto-generated string. See "Getting Started" section.
productId *	String	Search for specific product id.
callbackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductInventory xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <productId>prod123</productId>
      <callbackURL></callbackURL>
    </getProductInventory>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetInventoryResponse xmlns="">
  <ProductID>prod123</ProductID>
  <Inventory>10</Inventory>
</GetInventoryResponse>
```

3.4. updateProductInventory

Description: Update the stock of a specific product.

Parameters:

Name	Type	Description
storeURL *	String	3dCart Store URL from which the information will be requested. i.e.: www.3dcart.com



userKey *	String	Auto-generated string. See “Getting Started” section.
productId *	String	Search for specific product id.
quantity *	Number	Stock quantity for the specified product.
replaceStock *	Boolean	Indicates if the stock should be replaced or incremented by the new quantity.
callbackURL	String	URL where the XML response will be posted.

* Required parameters

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <updateProductInventory xmlns="http://3dcart.com/">
      <storeId>www.3dcart.com</storeId>
      <userKey>12345678912345678912345678912345</userKey>
      <productId>prod123</productId>
      <quantity>50</quantity>
      <replaceStock>true</replaceStock>
      <callbackURL></callbackURL>
    </updateProductInventory>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<UpdateInventoryResponse xmlns="">
  <ProductID>prod123</ProductID>
  <NewInventory>50</NewInventory>
</UpdateInventoryResponse>
```

Working with the callbackURL parameter

The callbackURL parameter can be provided to any method of the 3dcart web service, and must have a valid URL format, followed by the page that will handle the post.

i.e.: <http://www.3dcart.com/3dcartapiposthandler.asp>

When this parameter is provided, the web service will post the result xml to the specified page and return the following result to the requester:

```
<?xml version="1.0" encoding="utf-8" ?>
<Response xmlns="">
<Description>XML Posted to:
http://www.3dcart.com/3dcartapiposthandler.asp</Description>
</Response>
```

The entire XML will be posted to the specified page, which has to be prepared to handle the information.

Here is a sample code of a page that handles the post and save the XML into a file:

```
<%@language=vbscript%>

<% '*****
'3dCart Shopping Cart System - Version: 3.0
'3dCart Store Web Service - Version: 1.0

'Sample script to handle the xml response from the web service using
'the callbackURL parameter.
!*****%>

<%
Dim strXml, objXmlDoc

'Receives the xml
strXml = request.Form

'Loads the XML into a xmlDoc object.
Set objXmlDoc = server.CreateObject("Microsoft.XMLDOM")
objXmlDoc.preserveWhiteSpace = true
objXmlDoc.loadxml(strXml)

'Once the XML is loaded, you can do anything with the information.
'In this sample script, the XML is being saved into the "text.xml" file.
objXmlDoc.save(server.MapPath("test.xml"))

Set objXmlDoc = nothing
%>
```

Web Services Errors

The following errors can be returned by the web service method:

Type	Code	Description
XML file error: Not well-formed	0	XML FILE BAD FORMED
	1	UserIp Node Not Found / UserKey Node Not Found
	2	UserKey Node Not Found / UserIp Node Not Found
	3	ProductID Node Not Found
	4	Quantity Node Not Found
	5	BatchSize Node Not Found
	6	StartNum Node Not Found
	7	InvoiceNum Node Not Found
	8	NewStatus Node Not Found
Login Error	9	Invalid Method
	16	Bad IP
	17	Bad Key
	18	API Settings Not Enabled
Data Error: Invalid Data Request	31	Invalid Product Id
	32	Invalid Quantity
	33	Invalid Batch
	34	Invalid Start Number
	35	Invalid Invoice Number
	36	Invalid Date
	37	Invalid Range Between Dates
	38	Invalid Order Start From
	39	Invalid Status
	40	Error posting to callbackurl
	41	Invalid customerData
No Data Found	42	Invalid action
	43	Invalid data
	46	Product Not Found
	47	Customer Not Found
	48	Order Status Not Found
	49	Order Not Found

Example of a login error (invalid user key):

```
<?xml version="1.0" encoding="utf-8" ?>
<Error xmlns="">
  <Id>17</Id>
  <Description>Login Error</Description>
</Error>
```

Common Issues

* You must be sure to use your secure URL when making API calls. If you have using our shared SSL it will be <https://domain.3dcartstores.com>. If you have a dedicated SSL you will want to use <https://www.domain.com>. If the secure URL is not used you will see the following error:

*Error trying to get data from the store. Technical description: The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel**

Make sure the API is enabled and you have the API User Key in the Settings – General – Store Modules area.

Check the API settings area in the Settings – General – Store Modules area to be sure these are correctly configured for the action you are looking to perform.

Also, here is a link to our sample project:

<http://www.3dcart.com/downloads/API-sample.zip>

End - You've reached the end of this guide.

3dCart has many other features not listed on this guide, once your store is up and running continue exploring the features located on the different areas of your Store Manager. If you have additional questions about any of the features you can read the full documentation at <http://help.3dcart.com>; or clicking on the Help Icon from your Store Manager left navigation bar.

Contacting 3dCart

3dCart's support team is ready and willing to help you. Feel free to contact us at any time and also visit our online knowledgebase and user forums that have many of your questions and issues already answered.

Web: <http://support.3dcart.com>
KB: <http://support.3dcart.com/kb/>
Forums: <http://forums.3dcart.com>
Email: support@3dcart.com
Phone: (800) 828-6650 x 3

Thank you for using 3dCart.

Sincerely,

3dCart Support Team
support@3dcart.com