

# 3dCart Shopping Cart Software V3.X

Advanced Store Web Services



## Introduction

The Store Web Service gives merchants access to 3dCart Stores information. With the Advanced Store Web Service APIs, you can build applications to perform any task against the store database.

**Attention:** The advanced API allows complete database manipulation. 3dCart doesn't assume any responsibility for any possible damage caused by incorrect use of this feature.

## Table of Contents

|  |   |
|--|---|
| Table of Contents .....                          | 2 |
| Getting Started.....                             | 3 |
| Web Services Methods .....                       | 4 |
| 1.runQuery.....                                  | 4 |
| Working with the callBackURL parameter .....     | 6 |
| Web Services Errors.....                         | 7 |
| End - You've reached the end of this guide. .... | 8 |
| Contacting 3dCart.....                           | 8 |

## Getting Started

In order to use the Advanced Store Web Services, you first need to configure the API Settings in the Online Store Manager (Settings -> General -> Store Modules -> 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.
- Enable Advanced API: Enable/Disable the use of the Advanced API.

The API can be accessed by the following URL: [http://api.3dcart.com/cart\\_advanced.asmx](http://api.3dcart.com/cart_advanced.asmx)

## Web Services Methods

### 1. runQuery

Description: Executes SQL queries against the store database.

Parameters:

| Name           | Type   | Description  |
|----------------|--------|--|
| storeUrl *     | String | 3dCart Store URL from which the information will be requested.<br>i.e.: www.3dcart.com |
| userKey *      | String | Auto-generated string. See "Getting Started" section.                                  |
| sqlStatement * | String | SQL statement to run against the store database.                                       |
| callBackURL    | String | URL where the XML response will be posted.   |

\* Required parameters

Sample Request:

The following statement updates the product's price, increasing it by 15%.

```
<?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>
    <runQuery xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
      <sqlStatement>UPDATE products SET price = price * 1.15 WHERE catalogid =
123</sqlStatement>
      <callBackURL></callBackURL>
    </runQuery>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<runQueryResponse>1 affected records</runQueryResponse>
```

Running Multiple Queries:

The API runs multiple queries at once, just separate them by |;|.

The following example runs 2 selects and 1 update against the products table.

```
<?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>
    <runQuery xmlns="http://3dcart.com/">
      <storeUrl>www.3dcart.com</storeUrl>
      <userKey>12345678912345678912345678912345</userKey>
```

```
<sqlStatement>select top 1 catalogid, id from products|;;|update products set id =
'123' where catalogid = 1|;;|select top 1 catalogid, id from products</sqlStatement>
<callBackURL></callBackURL>
</runQuery>
</soap:Body>
</soap:Envelope>
```

## Sample Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<runQueryResponse>
  <runQueryResult>
    <runQueryResponse>
      <queryResult1>
        <runQueryRecord>
          <catalogid>1</catalogid>
          <id>test3</id>
        </runQueryRecord>
      </queryResult1>
      <queryResult2>1 affected records</queryResult2>
      <queryResult3>
        <runQueryRecord>
          <catalogid>1</catalogid>
          <id>123</id>
        </runQueryRecord>
      </queryResult3>
    </runQueryResponse>
  </runQueryResult>
</runQueryResponse>
```

## 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: 3.2
'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:<br>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                       |
|  | 9              | Invalid Method                                 |
| Login Error                            | 16             | Bad IP   |
|  | 17             | Bad Key  |
|  | 18             | API Settings Not Enabled                       |
| Data Error:<br>Invalid Data<br>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                           |
| 42                                     | Invalid action |  |
| 43                                     | Invalid data   |  |
| No Data Found                          | 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>
```

# 3dCart Shopping Cart Software V3.X

Advanced Store Web Services



## 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](mailto:support@3dcart.com)

Phone: (800) 828-6650 x 3

Thank you for using 3dCart.

Sincerely,

3dCart Support Team

[support@3dcart.com](mailto:support@3dcart.com)