

Cart Quote

The complete Magento Quotation manager

Manual Appendix I



Version	1.0.1
Support questions:	support@cart2quote.com
Sales questions:	sales@cart2quote.com
Website:	www.cart2quote.com
Release Date:	09/10/2015

Cart2Quote webservice

Cart2Quote Enterprise Edition is often used to communicate quote requests and proposals to and from other systems such as ERP and CRM software. This is possible using Cart2Quote Enterprise Edition because it has built-in webservices. The following section describes how to utilize these webservices.



Important: you need Cart2Quote Enterprise Edition to make use of Cart2Quote's webservices. Purchase [Cart2Quote Enterprise](#) or [Update your Cart2Quote version](#).

Below you will find how to retrieve quote information and how to make a proposal based on the quote information. Every section has some small example code.

1.1 Preparations

You will need to create a webservice user and role that can make use of the webservice resources of Cart2Quote. Please see this guide if you do not know how:

<http://www.richardcarterconsultancy.com/2012/06/creating-a-magento-api-user-account-in-magento-1-6/>

In pseudo PHP code, the following is always needed when communicating with Cart2Quote webservices:

```
$url = YOUR STORE'S URL; //'http://www.yourstore.com';  
define("SOAP_WSDL", $url . '/index.php/api/?wsdl');  
define("SOAP_WSDL2", $url . '/index.php/api/v2_soap/?wsdl');  
define("SOAP_USER", YOUR_USER_NAME);  
define("SOAP_PASS", YOUR_PASSWORD);  
$client = new SoapClient(SOAP_WSDL);  
$session = $client->login(SOAP_USER, SOAP_PASS);
```

You can view the available SOAP methods through the supplied WSDL. This document describes and provides sample usages of the most important methods.

1.2 Retrieving quote information

Cart2Quote follows the same way as Magento's webservices for information retrieval, by letting you retrieve lists based on filters, and by letting you ask for details about a single quote request.



1.2.1 Retrieving lists

c2q_quotation.list is the method that retrieves a list of quote requests. It can take an array as its argument which can take several different filters, including "status". In the example below, status 53 is a pending quote request. Listing all available filters and statuses is still pending for this document. If you need to know how to apply a certain filter to the quotation list, please contact [Cart2Quote support](#).

```
$result = $client->call($session, 'c2q_quotation.list');  
$result = $client->call($session, 'c2q_quotation.list', array(array('status' => 53)));  
print_r($result);
```

1.2.2 Retrieving detailed quote request information

c2q_quotation.info is the method that retrieves detailed quotation information. It has a quote ID as its argument.

```
$result = $client->call($session, 'c2q_quotation.info', array('quote_id' => $quote_id));  
print_r($result);
```

1.3 Making a proposal

Once a quote request is in Cart2Quote's system, sending out a proposal has a number of optional steps. The only required step to send out a proposal is to tell Cart2Quote to actually send out a proposal. The following actions can be done for a proposal, using webservice:

- adding quantities with according prices (OPTIONAL)
- modify a price for a certain quantity of a specific product (OPTIONAL)
- removing quantities (OPTIONAL)
- set shipping type and cost (OPTIONAL)
- set a comment for an item in the proposal (OPTIONAL)
- set a comment for the entire proposal (OPTIONAL)
- delete a product (OPTIONAL)
- send proposal (REQUIRED)



1.3.1 Adding quantities with according prices (OPTIONAL)

c2q_quotation.add_qtybyitem is the method that adds a quantity to an item in the quote request. This method needs an array of data as in the example below.

The first thing you will need is a product ID. You can find product IDs inside the proposal by looping through the 'totalRecords' index of the result array of a detailed quote info call.

```
$result = $client->call($session, 'c2q_quotation.info', array('quote_id' => $quote_id));
if( (isset($result['totalRecords']) && $result['totalRecords']>0) && isset($result['items']) ) {
    $flag_errors=0;
    foreach($result['items'] as $key=>$row) {
        $id      = $row['primary_key'];
        $product_id = $row['product_id'];
```

Next, you can add a quantity as follows:

```
$data = array(
    'quote_id'      => $quote_id,
    'product_id'   => $product_id,
    'request_qty'  => 15, //the quantity to add
    'owner_base_price'=> 100,
    'original_price' => 100,
    'quoteadv_product_id'=>$id //primary key of result row, see previous code block
);
$result = $client->call($session, 'c2q_quotation.add_qtybyitem', array($data));
```



1.3.2 Modify a price for a certain quantity of a specific product (OPTIONAL)

c2q_quotation.modify_requested_qty modifies an existing quantity. The webservice does not allow to add the same quantity twice, if you need to change the price for a quantity this method can be called. It has the same input array as the call to add a quantity from the previous paragraph.

```
$result = $client->call($session, 'c2q_quotation.modify_requested_qty', array($data));
```

1.3.3 removing quantities (OPTIONAL)

'c2q_quotation.delete_requested_qty' removes a quantity for a product in the proposal. It has an array as input and requires a request_id which is taken from the quotation.info call.

```
$data = array(
    'request_id' => 64, ///take data from 'c2q_quotation.info' -> for each requested qty you can retrieve
    unique 'request_id' value
    'quote_id' => $quote_id
);
$result = $client->call($session, 'c2q_quotation.delete_requested_qty', array($data));
```

1.3.4 Set shipping type and cost (OPTIONAL)

c2q_quotation.set_shipping sets shipping prices.

Cart2Quote has three different options for shipping price calculation:

- 0: use Magento's default
- 1: flat price per product
- 2: price for entire proposal

This is reflected in the webservice. Call the service using an array with quote id, shipping price and shipping type.

Example:

```
$data = array(
    'quote_id' => $quote_id, ///gotten from quotation.list
    'shipping_price' => 111,
    'shipping_type' => "0" ///options are 0, 1 and 2, see above
);
$result = $client->call($session, 'c2q_quotation.set_shipping', array($data));
```



1.3.5 Set a comment for an item in the proposal (OPTIONAL)

c2q_quotation.set_item_comment places a comment for a certain item in the proposal. It needs an array with product id and comment. The product ID can be retrieved by calling the quotation.info webservice, looping over the 'items' index of the result and getting the 'primary_key' indexes of those rows:

```
foreach($result['items'] as $key=>$row) {
    $id      = $row['primary_key'];
    $data = array(
        'quoteadv_product_id' => $id,
        'comment' => "This is a product comment"
    );
    $result = $client->call($session, 'c2q_quotation.set_item_comment', array($data));
}
```

1.3.6 Set a comment for the entire proposal (OPTIONAL)

c2q_quotation.set_proposal_comment sets a general comment for the entire proposal. It needs an array with quote id (from quotation.list) and a comment.

```
$data = array(
    'quote_id' => $quote_id,
    'comment' => "This is a proposal comment."
);
$result = $client->call($session, 'c2q_quotation.set_proposal_comment', array($data));
```

1.3.7 Delete a product (OPTIONAL)

c2q_quotation.delete_requested_item removes an item (product) from the quotation.

Example:

```
$data = array(
    'primary_key' => 23, //take data from 'c2q_quotation.info'
    'quote_id' => $quote_id
);
$result = $client->call($session, 'c2q_quotation.delete_requested_item', array($data));
```

1.3.8 Send proposal (REQUIRED)

c2q_quotation.send_proposal is the most important method, sending the proposal! A proposal can be sent at any time, all the above is optional. Usage is simple, just supply the quote ID (retrieved from quotation list call).

```
$result = $client->call($session, 'c2q_quotation.send_proposal', $quote_id);
```

