



2CHECKOUT.COM

Advanced User Guide

Table of Contents

Introduction.....	4
Purchase Routine.....	4
Parameter Sets	5
Plug-and-Play Parameters.....	5
Pass-Through-Products Parameters	9
Third Party Cart Parameters.....	14
Authorize.net Parameter Set.....	18
Return Process.....	22
MD5 Hash.....	24
Approved URL Passback.....	24
INS Notifications	25
Sale Notifications	26
Messages Supported & INS Message Parameters.....	26
Order Created.....	26
Fraud Status Changed	29
Shipping Status Changed.....	32
Invoice Status Changed	35
Refund Issued	38
Recurring Installment Success.....	41
Recurring Installment Failed.....	44
Recurring Stopped	47
Recurring Complete	50
Recurring Restarted	52
API.....	55
acct/detail_company_info	55
acct/detail_contact_info	57
acct/detail_pending_payment	59
acct/list_payments.....	61
sales/detail_sale	62
sales/list_sales.....	72
sales/refund_invoice.....	74
sales/refund_lineitem.....	76
sales/stop_lineitem_recurring	78
sales/mark_shipped.....	79
sales/create_comment	80
products/detail_product.....	81
products/list_products.....	85
products/create_product.....	89
products/update_product.....	91
products/delete_product.....	93
products/detail_option	94
products/list_options.....	96
products/create_option	99
products/update_option.....	100
products/delete_option	101
products/detail_coupon.....	102
products/list_coupons	103

products/create_coupon105
products/update_coupon106
products/delete_coupon107
Errors.....108
Conclusion109

INTRODUCTION

The following guide discusses our more advanced integration options and features that we offer. This user guide was written for cart developers and sellers who are developing their own custom integration. If you are interested in using our Plug-and-Play cart or are using a third party cart application, please visit our User Guide for integration instructions. Many account related instructions in this guide are written with the assumption that you have already signed up for your 2Checkout seller account and have already logged in at: <https://www.2checkout.com/va/>

If you are developing a third party cart application and require a developer account you can obtain one by signing up for a free demo account at the following URL: <https://www.2checkout.com/va/>. Once you have completed the sign up process you will receive a Seller/Client ID. Please email this ID to techsupport@2co.com so that we can update your developer account accordingly. Please Note: Developer accounts can never be paid out so any live test sales will need to be canceled immediately after being placed.

PURCHASE ROUTINE

The first step in integrating with 2Checkout is deciding which purchase routine will be better suited for your site. Currently, we offer 2 separate purchase routines which are both offered in our 15 supported languages and 30 supported currencies. Each purchase routine has its own advantages and requirements that are listed below. Please Note: The look and feel of our purchase routines cannot be altered or customized. If you would like to add your company logo to the purchase pages, you can find instructions in our User Guide.

Standard Purchase Routine

<https://www.2checkout.com/checkout/purchase>

Our standard purchase routine can be used with any parameter set and offers all of our payment options (if applicable) to the customer. These payment options include Credit Card (VISA, MasterCard, American Express, Discover/Novus, Diners Club and JCB, all derivatives of these brands are also supported), PayPal Express Checkout and PIN Debit. When using this purchase routine, the customer will be required to navigate through several pages and fill in the requested billing and shipping information (if applicable) to complete their order. An example of this purchase routine process can be viewed by clicking on the link below:

https://www.2checkout.com/checkout/spurchase?quantity=1&product_id=1&sid=1303908

Single Page Checkout

<https://www.2checkout.com/checkout/spurchase>

Our single page checkout can be used with any parameter set and also offers all of our payment options (if applicable) to the customer. These payment options include Credit Card (VISA, MasterCard, American Express, Discover/Novus, Diners Club and JCB, all derivatives of these brands are also supported), PayPal Express Checkout and PIN Debit. This purchase routine will list all of the customer's cart information and billing information fields on a single page. An example of this purchase routine process can be viewed by clicking on the link below:

https://www.2checkout.com/checkout/spurchase?quantity=1&product_id=4&sid=1303908

Please Note: Since the single page checkout does not ask for shipping information, any tangible orders passed into the single page checkout with our Plug-and-Play or Pass-Through-Products parameter sets will be reverted to our standard purchase routine. Orders passed in using either our third party cart or Authorize.net parameter sets will load properly in our single page purchase routine regardless of what the customer is ordering.

PARAMETER SETS

The next step in integrating with 2Checkout is getting your site to pass orders into one of our purchase routines correctly. Since our purchase routines work off of a Common Gateway Interface (CGI), your site will need to contain an HTML form or link with CGI parameters appended that passes the correct parameters (and the customer) to one of our purchase routines for checkout. Passing in the proper parameters will result in the customer going through the rest of our checkout process (such as entering their credit card information), and after verification, being returned back to your site with a sale confirmation. We support 4 different parameter sets which when passed into our purchase routine correctly, will create a sale. You may pass in any additional parameters that you may need and they will be returned to you at the end of the sale. The only restrictions on custom parameters are that they cannot share the name of any parameter that our system uses. Please Note: You will need a return script set up at your approved URL to receive any of these parameters back as they are not included in the confirmation emails. If you would like to test your integration of these parameters, please visit developers.2checkout.com. Please Note: You cannot mix the parameter sets. This will cause problems during payment or may prevent the payment pages from working at all.

PLUG-AND-PLAY PARAMETERS

These parameters can only be used if you have added your products to the 2CO database. This parameter supports both recurring products and also tangible products that will apply shipping charges based on the shipping methods you setup in your seller account.

Required Parameters:

`sid`

Your 2Checkout account number. (64 characters max)

`product_id`

The 2CO ID for the product or service in your 2Checkout account.

`quantity`

The quantity of the product. (99 max value)

Additional Parameters:

`demo`

Y to enable demo mode. Do not pass this in for live sales.

`fixed`

Y to remove the Continue Shopping button and lock the quantity fields.

`lang`

Chinese – `zh`, Danish – `da`, Dutch – `nl`, French – `fr`, German – `gr`, Greek – `el`, Italian – `it`, Japanese – `jp`, Norwegian – `no`, Portuguese – `pt`, Slovenian – `sl`, Spanish – `es_ib`, Spanish – `es_la`, Swedish – `sv`, defaults to English if this is absent, but `en` may be used for English as well.

`return_url`

Used to control where the Continue Shopping button will send the customer when clicked. (255 characters max)

`merchant_order_id`

Specify your order number with this parameter. It will also be included in the confirmation emails to yourself and the customer. (50 characters max)

`pay_method`

CC for Credit Card, PPI for PayPal. This will set the default selection on the payment method step during the checkout process.

`skip_landing`

If set to 1 it will skip the order review page of the purchase routine. If there are options on the products it will cause an error and redirect the customer back to the order review page.

`x_receipt_link_url`

Used to specify an approved URL on-the-fly, but is limited to the same domain that is used for your 2Checkout account, otherwise it will fail. This parameter will over-ride any URL set on the Site Management page. With the return method set to Given links back to my website, this parameter will control where the “Click Here To Finalize Your Order” button on the final page takes the customer, but the return URLs on the products will be listed as links below the button, next to the description of the product. If the return method is set to Direct Return or Header Redirect and one product is ordered, this parameter will NOT over-ride the return URLs on the products. If the return method is set to Direct Return or Header Redirect and more than one product is ordered, then this parameter will over-ride the product return URLs. Also keep in mind that many third party shopping carts recommend that the return method be set to Given links back to my website to work properly.

`coupon`

Specify a 2Checkout created coupon code. If applicable, the coupon will be automatically applied to the sale.

To populate the billing information page you may pass in:

`card_holder_name`

Card holder’s name. (128 characters max) The card holder’s name can also be populated using the

`first_name`, `middle_initial`, and `last_name` parameters. IMPORTANT: If you use these three

parameters, the data will be combined into and returned as `card_holder_name` to your Approved URL script.

`street_address`

Card holder's street address. (64 characters max)

`street_address2`

The second line for the street address, typically suburb or apartment number information. (64 characters max)

`city`

Card holder's city. (64 characters max)

`state`

Card holder's state. (64 characters max)

`zip`

Card holder's zip or postal code. (16 characters max)

`country`

Card holder's country. (64 characters max)

`email`

Card holder's email address. (64 characters max)

`phone`

Card holder's phone number. (16 characters max)

`phone_extension`

Card holder's phone extension. (9 characters max)

To populate the shipping information page you may pass in:

`ship_name`

Recipient's name. (128 characters max)

`ship_street_address`

Recipient's street address. (64 characters max)

`ship_street_address2`

The second line for the street address, typically suburb or apartment number information. (64 characters max)

`ship_city`

Recipient's city. (64 characters max)

`ship_state`

Recipient's state. (64 characters max)

`ship_zip`

Recipient's zip. (16 characters max)

`ship_country`

Recipient's country. (64 characters max)

PLEASE NOTE: Shipping information will only be collected for items that are defined as tangible in our system. Shipping is also only charged for tangible items.

Parameters Returned on a Successful Sale

`card_holder_name`

Provides the customer's name.

`city`

Provides the customer's city.

`country`

Provides the customer's country.

`credit_card_processed`

This parameter will always be passed back as Y. Can be used to confirm the funds for the order are available.

`demo`

Defines if an order was live, or if the order was a demo order. If the order was a demo, the MD5 hash will fail.

email
Provides the email address the customer provided when placing the order.

fixed
This parameter will only be passed back if it was passed into the purchase routine.

ip_country
Provides the customer's IP location. Useful if you perform your own additional fraud review.

key
An MD5 hash used to confirm the validity of a sale. It is calculated based on a combination of your secret word, seller identification number (sid), the order number, and the sale total.

lang
Advises the language the customer was able to view the Order Details page in. Can be used to track what language your customers speak or read. Can be helpful to track which pages or purchase buttons customers are using if you have multiple languages on your site.

merchant_order_id
The order ID you had assigned to the order. This parameter will only be passed back if it was passed into the purchase routine.

merchant_product_id
The ID you had assigned to your product.

order_number
The 2Checkout order number associated with the order.

invoice_id
The 2Checkout invoice number.

pay_method
Provides seller with the customer's payment method. CC for Credit Card, PPI for PayPal.

phone
Provides the phone number the customer provided when placing the order.

product_description
The description you had provided for your product.

product_id
The ID you had provided for your product.

quantity
The quantity of the product that was ordered.

ship_name
Provides the ship to name for the order.

ship_street_address
Provides ship to address.

ship_street_address2
Provides more detailed shipping address if this information was provided by the customer.

ship_city
Provides ship to city.

ship_state
Provides ship to state.

ship_zip
Provides ship to zip or postal code.

ship_country
Provides ship to country.

sid

The seller identification number. This can be useful if your return script is being used for multiple 2Checkout accounts. The `sid` parameter is also used to form the returned MD5 hash key to confirm the validity of a sale.

`state` Provides the state the customer gave when placing the order.

`street_address`

Provides the customer's street address.

`street_address2`

Provides more detailed address if more information is provided by the customer.

`total`

The amount the customer was billed on the order.

`zip`

Provides the customer's zip or postal code.

PLUG AND PLAY EXAMPLES:

Passing Multiple Plug-and-Play Products

You may also pass multiple Plug-and-Play products in one form. To do this, add a corresponding digit to the `quantity` and `product_id` parameters to group them for each product. For example, if we wanted to pass two products into the system we would be passing in first the `sid` parameter, then `product_id1` and `quantity1`, and then `product_id2` and `quantity2`.

Example:

```
<form action='https://www.2checkout.com/checkout/purchase' method='post'>
<input type='hidden' name='sid' value='1303908' />
<input type='hidden' name='quantity1' value='1' />
<input type='hidden' name='product_id1' value='1' />
<input type='hidden' name='quantity2' value='1' />
<input type='hidden' name='product_id2' value='3' />
<input type='hidden' name='quantity3' value='1' />
<input type='hidden' name='product_id3' value='4' />
<input name="submit" type='submit' value='Buy from 2CO' />
</form>
```

PASS-THROUGH-PRODUCTS PARAMETERS

These parameters can be used to dynamically pass in tangible/intangible and recurring/non-recurring products, product options, shipping taxes and coupons as items without having to create the products in the sellers area.

Required Parameters:

sid

Your 2Checkout account number. (64 characters max)

mode

Defines the parameter set. Should always be passed as '2CO'.

li_#_type

The type of line item that is being passed in. ('PRODUCT', 'SHIPPING', 'TAX' OR 'COUPON', DEFAULTS TO 'PRODUCT')

li_#_name

Name of the item passed in with the corresponding li_#_type. (128 characters max, cannot use '<' or '>', defaults to capitalized version of 'type'.)

li_#_quantity

Quantity of the item passed in with the corresponding li_#_type. (1-999, DEFAULTS TO 1 IF NOT PASSED IN.)

li_#_price

Price of the line item. Format: 0.00-99999999.99, defaults to 0 if a value isn't passed in, no negatives (use positive values for coupons), leading 0 & decimal are optional. *Important note: If the li_#_price parameter isn't used to pass in line item pricing, the pricing for the corresponding item will default to 0.00.*

li_#_tangible

Specifies if the corresponding li_#_type is a tangible or intangible. 'Y' OR 'N', if li_#_type is 'shipping' forced to 'Y'.

You can use these parameters to pass in multiple lineitems as well. To do so you will need to append a digit to each parameter. For example, li_0_type, li_0_quantity, li_0_name, li_0_price, li_0_product_id, li_0_description, li_0_tangible will describe the first product, then to describe the second product you would use a 1 so the parameters will read li_1_type, li_1_quantity, and so on. **Please note:** With this parameter set # may be represented by 0-999 and aren't required to be consecutive.

Additional Parameters:

li_#_product_id

Id that you use to identify the product. 64 CHARACTERS MAX, CANNOT USE '<' OR '>'.

li_#_product_description

Description you use for the product. 255 CHARACTERS MAX, CANNOT USE '<' OR '>'.

li_#_recurrence

WEEK | MONTH | YEAR – ALWAYS SINGULAR.

li_#_duration

FOREVER OR # WEEK | MONTH | YEAR – ALWAYS SINGULAR, DEFAULTS TO FOREVER.

li_#_startup_fee

AMOUNT IN ACCOUNT PRICING CURRENCY.

li_#_option_#_name

NAME OF OPTION. 64 CHARACTERS MAX – CANNOT INCLUDE '<' OR '>'.

li_#_option_#_value

NAME OF OPTION. 64 CHARACTERS MAX – CANNOT INCLUDE '<' OR '>'.

li_#_option_#_surcharge

AMOUNT IN ACCOUNT PRICING CURRENCY.

demo

Y to enable demo mode. Do not pass this in for live sales.

fixed

Y to remove the Continue Shopping button and lock the quantity fields.

`lang`

Chinese – `zh`, Danish – `da`, Dutch – `nl`, French – `fr`, German – `gr`, Greek – `el`, Italian – `it`, Japanese – `jp`, Norwegian – `no`, Portuguese – `pt`, Slovenian – `sl`, Spanish – `es_ib`, Spanish – `es_la`, Swedish – `sv`, defaults to English if this is absent, but `en` may be used for English as well.

`return_url`

Used to control where the Continue Shopping button will send the customer when clicked. (255 characters max)

`merchant_order_id`

Specify your order number with this parameter. It will also be included in the confirmation emails to yourself and the customer. (50 characters max)

`pay_method`

CC for Credit Card, PPI for PayPal. This will set the default selection on the payment method step during the checkout process.

`skip_landing`

If set to 1 it will skip the order review page of the purchase routine. If there are options on the products it will cause an error and redirect the customer back to the order review page.

`x_receipt_link_url`

Used to specify an approved URL on-the-fly, but is limited to the same domain that is used for your 2Checkout account, otherwise it will fail. This parameter will over-ride any URL set on the Site Management page. With the return method set to Given links back to my website, this parameter will control where the “Click Here To Finalize Your Order” button on the final page takes the customer, but the return URLs on the products will be listed as links below the button, next to the description of the product. If the return method is set to Direct Return or Header Redirect and one product is ordered, this parameter will NOT over-ride the return URLs on the products. If the return method is set to Direct Return or Header Redirect and more than one product is ordered, then this parameter will over-ride the product return URLs. Also keep in mind that many third party shopping carts recommend that the return method be set to Given links back to my website to work properly.

`coupon`

Specify a 2Checkout created coupon code. If applicable, the coupon will be automatically applied to the sale.

To populate the billing information page you may pass in:

`card_holder_name`

Card holder’s name. (128 characters max) The card holder’s name can also be populated using the `first_name`, `middle_initial`, and `last_name` parameters. IMPORTANT: If you use these three parameters, the data will be combined into and returned as `card_holder_name` to your Approved URL script.

`street_address`

Card holder’s street address. (64 characters max)

`street_address2`

The second line for the street address, typically suburb or apartment number information. (64 characters max)

`city`

Card holder’s city. (64 characters max)

`state`

Card holder’s state. (64 characters max)

`zip`

Card holder’s zip or postal code. (16 characters max)

`country`

Card holder’s country. (64 characters max)

email
Card holder's email address. (64 characters max)

phone
Card holder's phone number. (16 characters max)

phone_extension
Card holder's phone extension. (9 characters max)

To populate the shipping information page you may pass in:

ship_name
Recipient's name. (128 characters max)

ship_street_address
Recipient's street address. (64 characters max)

ship_street_address2
The second line for the street address, typically suburb or apartment number information. (64 characters max)

ship_city
Recipient's city. (64 characters max)

ship_state
Recipient's state. (64 characters max)

ship_zip
Recipient's zip. (16 characters max)

ship_country
Recipient's country. (64 characters max)

Please Note: Shipping information will only be collected for items that are defined as tangible in our system. Shipping is also only charged for tangible items.

Parameters Returned on a Successful Sale

card_holder_name
Provides the customer's name.

city
Provides the customer's city.

country
Provides the customer's country.

credit_card_processed
This parameter will always be passed back as Y. Can be used to confirm the funds for the order are available.

demo
Defines if an order was live, or if the order was a demo order. If the order was a demo, the MD5 hash will fail.

email
Provides the email address the customer provided when placing the order.

fixed
This parameter will only be passed back if it was passed into the purchase routine.

ip_country
Provides the customer's IP location. Useful if you perform your own additional fraud review.

key
An MD5 hash used to confirm the validity of a sale. It is calculated based on a combination of your secret word, seller identification number (sid), the order number, and the sale total.

lang

Advises the language the customer was able to view the Order Details page in. Can be used to track what language your customers speak or read. Can be helpful to track which pages or purchase buttons customers are using if you have multiple languages on your site.

li_#_type

The type of the corresponding lineitem.

li_#_name

Name of the corresponding lineitem.

li_#_quantity

Quantity of the corresponding lineitem.

li_#_price

Price of the corresponding lineitem.

li_#_tangible

Specifies if the corresponding li_#_type is a tangible or intangible. 'Y' OR 'N'

li_#_product_id

ID of the corresponding lineitem.

li_#_product_description

Description of the corresponding lineitem.

li_#_recurrence

WEEK | MONTH | YEAR – always singular.

li_#_duration

Forever or # WEEK | MONTH | YEAR – always singular, defaults to Forever.

li_#_startup_fee

Amount in account pricing currency.

li_#_option_#_name

Name of option. 64 characters max – cannot include '<' or '>'.

li_#_option_#_value

Name of option. 64 characters max – cannot include '<' or '>'.

li_#_option_#_surcharge

Amount in account pricing currency.

merchant_order_id

The order ID you had assigned to the order. This parameter will only be passed back if it was passed into the purchase routine.

order_number

The 2Checkout order number associated with the order.

invoice_id

The 2Checkout invoice number.

pay_method

Provides seller with the customer's payment method. CC for Credit Card, PPI for PayPal.

phone

Provides the phone number the customer provided when placing the order.

ship_name

Provides the ship to name for the order.

ship_street_address

Provides ship to address.

ship_street_address2

Provides more detailed shipping address if this information was provided by the customer.

ship_city

Provides ship to city.

ship_state

Provides ship to state.

ship_zip

Provides ship to zip or postal code.

ship_country

Provides ship to country.

sid

The seller identification number. This can be useful if your return script is being used for multiple 2Checkout accounts. The sid parameter is also used to form the returned MD5 hash key to confirm the validity of a sale.

state Provides the state the customer gave when placing the order.

street_address

Provides the customers street address.

street_address2

Provides more detailed address if more information is provided by the customer.

total

The amount the customer was billed on the order.

zip

Provides the customers zip or postal code.

Please Note: If an order includes a product that is defined as tangible or a shipping method it will need to be marked as shipped before it will deposit into your pending payment. If the order is not marked as shipped within 7 days of being placed, it will need to be reauthorized prior to being marked as shipped. For more information on marking sales as shipped in the admin, please visit the Marking Orders as Shipped section of our User Guide. This can also be done programmatically using our reauth and mark_shipped API calls which are outlined in the API section of this document.

PASS-THROUGH-PRODUCT EXAMPLES:

Tangible Lineitem

```
<form action='https://www.2checkout.com/checkout/purchase' method='post'>
<input type='hidden' name='sid' value='1303908' >
<input type='hidden' name='mode' value='2CO' >
<input type='hidden' name='li_0_type' value='product' >
<input type='hidden' name='li_0_name' value='Example Product Name' >
<input type='hidden' name='li_0_price' value='1.00' >
<input type='hidden' name='li_0_quantity' value='1' >
<input type='hidden' name='li_0_tangible' value='Y' >
<input name='submit' type='submit' value='Buy from 2CO' >
</form>
```

Intangible Lineitem

```
<form action='https://www.2checkout.com/checkout/purchase' method='post'>
<input type='hidden' name='sid' value='1303908' >
<input type='hidden' name='mode' value='2CO' >
<input type='hidden' name='li_0_type' value='product' >
<input type='hidden' name='li_0_name' value='Example Product Name' >
<input type='hidden' name='li_0_price' value='1.00' >
<input type='hidden' name='li_0_quantity' value='1' >
<input name='submit' type='submit' value='Buy from 2CO' >
</form>
```

Recurring Lineitem

```
<form action='https://www.2checkout.com/checkout/purchase' method='post'>
<input type='hidden' name='sid' value='1303908' >
<input type='hidden' name='mode' value='2CO' >
<input type='hidden' name='li_0_type' value='product' >
<input type='hidden' name='li_0_name' value='Example Product Name' >
<input type='hidden' name='li_0_price' value='1.00' >
<input type='hidden' name='li_0_quantity' value='1' >
<input type='hidden' name='li_0_recurrence' value='1 Month' >
<input type='hidden' name='li_0_duration' value='Forever' >
<input name='submit' type='submit' value='Buy from 2CO' >
</form>
```

Product Options

```
<form action='https://www.2checkout.com/checkout/purchase' method='post'>
<input type='hidden' name='sid' value='1303908' >
<input type='hidden' name='mode' value='2CO' >
<input type='hidden' name='li_0_type' value='product' >
<input type='hidden' name='li_0_name' value='Example Product Name' >
<input type='hidden' name='li_0_price' value='1.00' >
<input type='hidden' name='li_0_quantity' value='1' >
<input type='hidden' name='li_0_tangible' value='Y' >
<input type='hidden' name='li_0_option_0_name' value='Product Option Name' >
<input type='hidden' name='li_0_option_0_value' value='Product Option Value' >
<input type='hidden' name='li_0_option_0_surcharge' value='1.00' >
<input name='submit' type='submit' value='Buy from 2CO' >
</form>
```


Product with a quantity 2, a coupon deduction, shipping and tax surcharges, billing and shipping information.

```
<form action='https://www.2checkout.com/checkout/purchase' method='post'>
<input type='hidden' name='sid' value='1303908' >
<input type='hidden' name='mode' value='2CO' >
<input type='hidden' name='li_0_type' value='product' >
<input type='hidden' name='li_0_name' value='Example Product Name' >
<input type='hidden' name='li_0_product_id' value='Example Product ID' >
<input type='hidden' name='li_0_description' value='Example Product Description' >
<input type='hidden' name='li_0_price' value='10.00' >
<input type='hidden' name='li_0_quantity' value='2' >
<input type='hidden' name='li_0_tangible' value='Y' >
<input type='hidden' name='li_1_type' value='shipping' >
<input type='hidden' name='li_1_name' value='Example Shipping Method' >
<input type='hidden' name='li_1_price' value='1.50' >
<input type='hidden' name='li_2_type' value='coupon' >
<input type='hidden' name='li_2_name' value='Example Coupon' >
<input type='hidden' name='li_2_price' value='1.00' >
<input type='hidden' name='li_3_type' value='tax' >
<input type='hidden' name='li_3_name' value='Example Tax' >
<input type='hidden' name='li_3_price' value='0.50' >
<input type='hidden' name='card_holder_name' value='Checkout Shopper' >
<input type='hidden' name='street_address' value='1785 OBrien Road' >
<input type='hidden' name='street_address2' value='Suite 200' >
<input type='hidden' name='city' value='Columbus' >
<input type='hidden' name='state' value='OH' >
<input type='hidden' name='zip' value='43228' >
<input type='hidden' name='country' value='USA' >
<input type='hidden' name='email' value='example@2co.com' >
<input type='hidden' name='phone' value='614-921-2450' >
<input type='hidden' name='phone_extension' value='197' >
<input type='hidden' name='ship_name' value='Gift Receiver' >
<input type='hidden' name='ship_street_address' value='1234 Address Road' >
<input type='hidden' name='ship_street_address2' value='Apartment 123' >
<input type='hidden' name='ship_city' value='Columbus' >
<input type='hidden' name='ship_state' value='OH' >
<input type='hidden' name='ship_zip' value='43235' >
<input type='hidden' name='ship_country' value='USA' >
<input name='submit' type='submit' value='Buy from 2CO' >
</form>
```

THIRD PARTY CART PARAMETERS

These are 2Checkout's proprietary set of parameters that you can use to pass third party cart sales into our purchase routine.

Required Parameters:

`sid`

Your 2Checkout account number.

`total`

The total amount to be billed, in decimal form, without a currency symbol or comma. (8 characters, decimal, 2 characters: Example: 99999999.99)

`cart_order_id`

A unique order ID from your program. (128 characters max)

Product Identification Parameters:

Sellers using this parameter set must pass in parameters to identify what is being sold during the purchase. These parameters will also benefit sellers as when used correctly the product information will be displayed to customers on the purchase pages and will be listed on the sellers 2Checkout.com sale detail page.

`id_type`

Due to changes in the Product Identification parameters this parameter should be passed in once during the purchase and will need to have its value set to 1.

`c_prod`

The ID of the product that is being purchased. The quantity of product being purchased can be passed with this parameter by placing a comma after the ID and then the quantity in numeric format.

`c_name`

The value of this parameter should contain the name of the product to be created. (128 characters max)

`c_description`

The value of this parameter should contain a short description of the product to be sold. (255 characters max)

`c_price`

The price of your product. Please note that in order to prevent a discrepancy between the total parameter and the sum of the product identification `c_price` parameters which may confuse the customer, the product price will not be displayed on the purchase page. (8 characters, decimal, 2 characters: Example: 99999999.99)

You can use these identification parameters to pass in multiple products as well. To do so you will need to append a digit to each parameter. For example, `c_prod_1`, `c_name_1`, `c_description_1`, `c_price_1` will describe the first product, then to describe the second product you would use a 2 so the parameters will read `c_prod_2`, `c_name_2` and so on.

Additional Parameters:

`demo`

Y to enable demo mode, do not pass this in for live sales.

`fixed`

Y to remove the Continue Shopping button and lock the quantity fields.

`lang`

Chinese – `zh`, Danish – `da`, Dutch – `nl`, French – `fr`, German – `gr`, Greek – `el`, Italian – `it`, Japanese – `jp`, Norwegian – `no`, Portuguese – `pt`, Slovenian – `sl`, Spanish – `es_ib`, Spanish – `es_la`, Swedish – `sv`, defaults to English if this is absent, but `en` may be used for English as well.

`return_url`

Used to control where the Continue Shopping button will send the customer when clicked. (255 characters max)

`merchant_order_id`

Specify your order number with this parameter. It will also be included in the confirmation emails to yourself and the customer. (50 characters max)

`pay_method`

CC for Credit Card, PPI for PayPal. This will set the default selection on the payment method step during the checkout process.

`skip_landing`

If set to 1 it will skip the order review page of the purchase routine.

`x_receipt_link_url`

Used to specify an approved URL on-the-fly, but is limited to the same domain that is used for your 2Checkout account, otherwise it will fail. This parameter will over-ride any URL set on the Site Management page.

`coupon`

Specify a 2Checkout created coupon code. If applicable, the coupon will be automatically applied to the sale.

To populate the billing information page you may pass in:

`card_holder_name`

Card holder's name. (128 characters max) The card holder's name can also be populated using the `first_name`, `middle_initial`, and `last_name` parameters. IMPORTANT: If you use these three parameters, the data will be combined into and returned as `card_holder_name` to your Approved URL script.

`street_address`

Card holder's street address. (64 characters max)

`street_address2`

The second line for the street address, typically suburb or apartment number information. (64 characters max)

`city`

Card holder's city. (64 characters max)

`state`

Card holder's state. (64 characters max)

`zip`

Card holder's zip or postal code. (16 characters max)

`country`

Card holder's country. (64 characters max)

`email`

Card holder's email address. (64 characters max)

`phone`

Card holder's phone number. (16 characters max)

`phone_extension`

Card holder's phone extension. (9 characters max)

To populate the shipping information page you may pass in:

`ship_name`

Recipient's name. (128 characters max)

`ship_street_address`

Recipient's street address. (64 characters max)

`ship_street_address2`

The second line for the street address, typically suburb or apartment number information. (64 characters max)

ship_city

Recipient's city. (64 characters max)

ship_state

Recipient's state. (64 characters max)

ship_zip

Recipient's zip or postal code. (16 characters max)

ship_country

Recipient's country. (64 characters max)

PLEASE NOTE: Additional shipping charges do not apply to third party cart sales.

Parameters Returned on a Successful Sale

card_holder_name

Provides the customer's name.

cart_id

The cart ID your cart assigned to the order.

cart_order_id

A unique order ID from your program that is normally used to identify an incomplete sale in your database so that it can be marked as paid.

city

Provides the customer's city.

country

Provides the customer's country.

credit_card_processed

This parameter will always be passed back as Y.

demo

Defines if an order was live, or if the order was a demo order. If the order was a demo, the MD5 hash will fail.

email

Provides the email address the customer provided when placing the order.

fixed

This parameter will only be passed back if it was passed into the purchase routine.

ip_country

Provides the customer's IP location. Useful if you perform your own additional fraud review.

key

An MD5 hash used to confirm the validity of a sale. It is calculated based on a combination of your secret word, seller identification number, the order number, and the sale total.

lang

Advises the language the customer was able to view the Order Details page in. Can be used to track what language your customers speak or read. Can be helpful to track which pages or purchase buttons customers are using if you have multiple languages on your site.

merchant_order_id

The order ID you assigned to the order. This parameter will only be passed back if it was passed into the purchase routine.

order_number

The 2Checkout order number associated with the order.

invoice_id

The 2Checkout invoice number.

pay_method

Provides seller with the customer's payment method. CC for Credit Card, PPI for PayPal.

phone
Provides the phone number the customer provided when placing the order.

ship_name
Provides the ship to name for the order.

ship_street_address
Provides ship to address.

ship_street_address2
Provides more detailed shipping address if more information is provided by the customer.

ship_city
Provides ship to city.

ship_state
Provides ship to state.

ship_zip
Provides ship to zip or postal code.

ship_country
Provides ship to country.

sid
The seller identification number. This can be useful if your return script might be in use for multiple 2Checkout accounts. The `sid` parameter is also used to form the returned MD5 hash key to confirm the validity of a sale.

state
Provides the state the customer gave when placing the order. Can be used for a customer database.

street_address
Provides the customers street address.

street_address2
Provides more detailed address if more information is provided by the customer.

total
The amount the customer was billed on the order.

zip
Provides the customers zip or postal code.

PLEASE NOTE: RECURRING BILLING CANNOT BE DONE WITH THIS PARAMETER SET. YOU MUST USE EITHER THE PLUG-AND-PLAY PARAMETER SET OR THE PASS-THROUGH-PRODUCTS PARAMETER SET TO PASS IN A RECURRING SALE.

THIRD PARTY CART EXAMPLES:

Passing multiple products, billing and shipping information using the single page checkout.

```
<form action='https://www.2checkout.com/checkout/spurchase' method='post'>
<input type='hidden' name='sid' value='1303908' >
<input type='hidden' name='total' value='3.00' >
<input type='hidden' name='cart_order_id' value='CART-001' >
<input type='hidden' name='id_type' value='1' >
<input type='hidden' name='c_prod_1' value='PRODUCT-10,1' >
<input type='hidden' name='c_name_1' value='Product 10' >
<input type='hidden' name='c_description_1' value='This is my 10th product' >
<input type='hidden' name='c_price_1' value='1.00' >
<input type='hidden' name='c_prod_2' value='PRODUCT-20,1' >
<input type='hidden' name='c_name_2' value='Product 20' >
<input type='hidden' name='c_description_2' value='This is my 20th product' >
<input type='hidden' name='c_price_2' value='2.00' >
<input type='hidden' name='card_holder_name' value='Checkout Shopper' >
<input type='hidden' name='street_address' value='1785 OBrien Road' >
<input type='hidden' name='street_address2' value='Suite 200' >
<input type='hidden' name='city' value='Columbus' >
<input type='hidden' name='state' value='OH' >
<input type='hidden' name='zip' value='43228' >
<input type='hidden' name='country' value='USA' >
<input type='hidden' name='email' value='example@2co.com' >
<input type='hidden' name='phone' value='614-921-2450' >
<input type='hidden' name='phone_extension' value='197' >
<input type='hidden' name='ship_name' value='Checkout Shopper' >
<input type='hidden' name='ship_street_address' value='1785 OBrien Road' >
<input type='hidden' name='ship_street_address2' value='Suite 200' >
<input type='hidden' name='ship_city' value='Columbus' >
<input type='hidden' name='ship_state' value='OH' >
<input type='hidden' name='ship_zip' value='43228' >
<input type='hidden' name='ship_country' value='USA' >
<input name='submit' type='submit' value='Buy from 2CO' >
</form>
```

AUTHORIZE.NET PARAMETER SET

The Authorize.net parameters that we support can be used to pass third party cart sales into our purchase routine.

Required Parameters:

`x_login`

Your numerical 2Checkout account number.

`x_amount`

The total amount to be billed, without a currency symbol or comma, and only two decimal places if used. (8 characters, decimal, 2 characters: Example: 99999999.99)

`x_invoice_num`

A unique order ID from your system. (64 characters max)

Product Identification Parameters:

Sellers using this parameter set must pass in parameters to identify what is being sold during the purchase. These parameters will also benefit sellers as when used correctly the product information will be displayed to customers on the purchase pages and will be listed on the sellers 2Checkout.com sale detail page.

`id_type`

Due to changes in with the Product Identification parameters this parameter should be passed in once during the purchase and will need to have its value set to 1.

`c_prod`

The ID of the product that is being purchased. The quantity of product being purchased can be passed with this parameter by placing a comma after the ID and then the quantity in numeric format.

`c_name`

The value of this parameter should contain the name of the product to be created. (128 characters max)

`c_description`

The value of this parameter should contain a short description of the product to be sold. (255 characters max)

`c_price`

The price of your product. Please note that in order to prevent a discrepancy between the total parameter and the sum of the product identification `c_price` parameters which may confuse the customer, the product price will not be displayed on the purchase page. (8 characters, decimal, 2 characters: Example: 99999999.99)

You can use these identification parameters to pass in multiple products as well. To do so you will need to append a digit to each parameter. For example, `c_prod_1`, `c_name_1`, `c_description_1`, `c_price_1` will describe the first product, than to describe the second product you would use a 2 so the parameters will read `c_prod_2`, `c_name_2` and so on.

Additional Parameters:

`demo`

Y to enable demo mode, do not pass this in for live sales.

`fixed`

Y to remove the Continue Shopping button and lock the quantity fields.

`lang`

Chinese – zh, Danish – da, Dutch – nl, French – fr, German – gr, Greek – el, Italian – it, Japanese – jp, Norwegian – no, Portuguese – pt, Slovenian – sl, Spanish – es_ib, Spanish – es_la, Swedish – sv, defaults to English if this is absent, but en may be used for English as well.

`return_url`

Used to control where the Continue Shopping button will send the customer when clicked. (255 characters max)

`merchant_order_id`

Specify your order number with this parameter. It will also be included in the confirmation emails to yourself and the customer. (50 characters max)

`pay_method`

CC for Credit Card, PPI for PayPal. This will set the default selection on the payment method step during the checkout process.

`skip_landing`

If set to 1 it will skip the order review page of the purchase routine.

`x_receipt_link_url`

Used to specify an approved URL on-the-fly, but is limited to the same domain that is used for your 2Checkout account, otherwise it will fail. This parameter will over-ride any URL set on the Site Management page.

`coupon`

Specify a 2Checkout created coupon code. If applicable, the coupon will be automatically applied to the sale.

To populate the billing information page you may pass in:

`x_first_name`

Buyers first name. (64 characters max)

`x_last_name`

Buyers last name. (64 characters max)

`x_address`

Card holder's street address. (64 characters max)

`x_city`

Card holder's city. (64 characters max)

`x_state`

Card holder's state. (64 characters max)

`x_zip`

Card holder's zip or postal code. (16 characters max)

`x_country`

Card holder's country. (64 characters max)

`x_email`

Card holder's email address. (64 characters max)

`x_phone`

Card holder's phone number. (16 characters max)

To populate the shipping information page you may pass in:

`x_ship_to_first_name`

Recipient's first name. (64 characters max)

`x_ship_to_last_name`

Recipient's last name. (64 characters max)

`x_ship_to_address`

Recipient's street address. (64 characters max)

`x_ship_to_city`

Recipient's city. (64 characters max)

`x_ship_to_state`

Recipient's state. (64 characters max)

`x_ship_to_zip`

Recipient's zip. (16 characters max)

`x_ship_to_country`

Recipient's country. (64 characters max)

Please Note: Additional shipping charges do not apply to third party cart sales.

Parameters Returned on a Successful Sale

`card_holder_name`

Provides the customer's name.

`demo`

Defines if an order was live, or if the order was a demo order. If the order was a demo, the MD5 hash will fail.

`fixed`

This parameter will only be passed back if it was passed into the purchase routine.

`ip_country`

Provides the customer's IP location. Useful if you perform your own additional fraud review.

`lang`

Advises the language the customer was able to view the Order Details page in. Can be used to track what language your customers speak or read. Can be helpful to track which pages or purchase buttons customers are using if you have multiple languages on your site.

`merchant_order_id`

The order ID you assigned to the order. This parameter will only be passed back if it was passed into the purchase routine.

`order_number`

The 2Checkout order number associated with the order.

`pay_method`

Provides seller with the customer's payment method. CC for Credit Card, PPI for PayPal.

`x_2checked`

This parameter will always be passed back as Y.

`x_address`

Provides the customer's street address.

`x_city`

Provides the customer's city.

`x_country`

Provides the customer's country.

`x_email`

Provides the email address the customer provided when placing the order.

`x_login`

The seller identification number. This can be useful if your return script might be in use for multiple 2Checkout accounts. The `x_login` parameter is also used to form the returned MD5 hash key to confirm the validity of a sale.

`x_md5_hash`

An MD5 hash used to confirm the validity of a sale. It is calculated based on a combination of your secret word, seller identification number, the order number, and the sale total.

`x_phone`

Provides the phone number the customer provided when placing the order.

`x_ship_to_first_name`

Provides the ship to name for the order.

`x_ship_to_last_name`

Provides the ship to name for the order.

`x_ship_to_address`

Provides ship to address.

`x_ship_to_city`

Provides ship to city.

`x_ship_to_country`

Provides ship to country.

x_ship_to_state
Provides ship to state.

x_ship_to_zip
Provides ship to zip or postal code.

x_state
Provides the state the customer gave when placing the order.

x_zip
Provides the customer's zip or postal code.

x_amount
The amount the customer was billed on the order.

x_invoice_num
A unique order ID from your cart that is normally used to identify an incomplete sale in your database so that it can be marked as paid.

x_response_code
Defaults to 1. Used by Authorize.net.

invoice_id
The 2Checkout invoice number.

x_trans_id
The 2Checkout order number associated with the order.

Please Note: Recurring billing cannot be done with this parameter set. You must use either the Plug-and-Play parameter set or the Pass-Through-Products parameter set to pass in a recurring sale.

AUTHORIZE.NET EXAMPLES:

Passing multiple products, billing and shipping information using the single page checkout.

```
<form action='https://www.2checkout.com/checkout/spurchase' method='post'>
<input type='hidden' name='x_login' value='1303908' >
<input type='hidden' name='x_amount' value='3.00' >
<input type='hidden' name='x_invoice_num' value='CART-001' >
<input type='hidden' name='id_type' value='1' >
<input type='hidden' name='c_prod_1' value='PRODUCT-10,1' >
<input type='hidden' name='c_name_1' value='Product 10' >
<input type='hidden' name='c_description_1' value='This is my 10th product' >
<input type='hidden' name='c_price_1' value='1.00' >
<input type='hidden' name='c_prod_2' value='PRODUCT-20,1' >
<input type='hidden' name='c_name_2' value='Product 20' >
<input type='hidden' name='c_description_2' value='This is my 20th product' >
<input type='hidden' name='c_price_2' value='2.00' >
<input type='hidden' name='x_first_name' value='Testing' >
<input type='hidden' name='x_last_name' value='Tester' >
<input type='hidden' name='x_address' value='1234 Address Road' >
<input type='hidden' name='x_city' value='Columbus' >
<input type='hidden' name='x_state' value='OH' >
<input type='hidden' name='x_zip' value='43235' >
<input type='hidden' name='x_country' value='USA' >
<input type='hidden' name='x_ship_to_first_name' value='Testing' >
<input type='hidden' name='x_ship_to_last_name' value='Tester' >
<input type='hidden' name='x_ship_to_address' value='1234 Address Road' >
<input type='hidden' name='x_ship_to_city' value='Columbus' >
<input type='hidden' name='x_ship_to_state' value='OH' >
<input type='hidden' name='x_ship_to_zip' value='43235' >
<input type='hidden' name='x_ship_to_country' value='USA' >
<input name='submit' type='submit' value='Buy from 2CO' >
</form>
```

RETURN PROCESS

When a sale has completed, we can return the customer and the sale parameters to a script or page on your site that you specify as the approved URL. You can set your account's approved URL on your account's Site Management by clicking on the **Account** tab and then selecting the **Site Management** sub category. Specifying an approved URL at the account level will direct all of your customers to the same page on your site when their sale has completed.

If you are using our Plug-and-Play parameter set the approved URL can also be set at the product level by setting up an approved URL for your products individually. Product level approved URLs have the ability to override the approved URL set at the account level and can direct the customer to a specific location for that product, such as a download page. If multiple products are ordered, we can provide the customer with multiple approved URLs so that they can access each product in the sale. Instructions on setting up your products can be found in the User Guide.

You may also choose to pass the approved URL on the fly by using the `x_receipt_link_url` parameter. This parameter will override both product and account level approved URLs but the value passed in must match the domain registered to the account.

If you are returning the customer to a script on your end it is important to note that parameter information will typically be returned by POST. Parameters however will be returned by GET if either the Header Redirect method is selected or the Plug-and-Play parameter set is used and multiple products are purchased at the same time. Thus if you are using a product level approved URL your return script should be able to handle both methods without breaking. You may also wish to verify the authenticity of each return post by using the MD5 hash that we pass back with each sale. More information on using this parameter and the MD5 hash can be found in the MD5 hash section of this document.

Finally, you will determine the method in which your customer is returned to the approved URL. 2Checkout.com provides three methods. You may send the customer to our order processed page which will display a Click Here to Finalize Your Order button to redirect the customer, you may bypass the order processed page using a header redirect or you can immediately display your approved URL to the customer while they remain on our server. Your return method can be selected on the Site Management page. With the Direct Return set to Given links back to my Website, the customer will be taken to our Order Processed page after completing a successful purchase. This page will feature a Click Here to Finalize Your Order button. When clicked the customer and collected sale parameters will be directed to the provided approved URL.

With Direct Return set to Direct Return (Your URL), sale parameters will be posted automatically to the approved URL while fetched by our server and displayed to the customer. When using this method the URL will be masked to the customer, appearing to still be on the 2Checkout.com domain. This method can be used with redirects as long as each page outputs content more than 255 characters to the browser. If this method encounters a page that redirects without outputting content the process will fail and the customer will be sent to our standard Order Processed page. This occurs usually with a header redirect, specifically content less than 255 characters. This is a common issue as many developers will set their approved URL to a script that processes the return sales parameters then silently forwards the customer to another page. This is usually a thank you or download page for intangible products. The best solution is to handle all post-order processing on the page set as your approved URL, including the thank you message. If no redirects are used, meaning the URL is masked by our servers, relative links will not point to the correct location. This can be corrected with the use of absolute paths on the approved URL page or by simply using a base tag in the head of the document to provide a reference for the relative paths.

With Direct Return set to Header Redirect the customer will be immediately returned to your approved URL. Using this method, the sale parameters will be returned along with the customer using the GET method. If you have problems with the return process you're welcome to contact us at techsupport@2co.com to assist with troubleshooting the issue. Please Note: If you do not specify an approved URL at the account level, product level, or with the `x_receipt_link_url` parameter the customer will remain on the 2Checkout Order Processed page upon completion of the order.***

MD5 HASH

The MD5 hash is provided to help you verify the authenticity of both the initial passback to your approved URL and all notifications sent by our Instant Notification Service. In this section, we will find out how the hash is calculated for both the initial passback to your approved URL and on INS notifications. Please Note: We intentionally break the hash code for demo orders by forcing the order number being computed to “1” so that you can compare the hash we provide with what it should be to determine whether or not to provide the customer with your goods. If you wish to test the hash on an approved URL passback, you’ll have to place a live test order using a real credit card number or amend your script to compute the hash for demo sales. Also, INS notifications are not sent on demo sales so you should use either a live sale or our INS simulator at developers.2checkout.com to test your INS script with the correct hash.

APPROVED URL PASSBACK

To calculate the MD5 hash on an approved URL passback, you need to make a string that contains the information described below and pass it in as the value to your scripting language's MD5 function. Below is an example:

```
md5 ( secret word + account number + order number + total )
```

The secret word is set by yourself on the Site Management page. The account number is your numerical seller/client ID number. The order number is the order number for the sale. The total is the numerical value for the total amount of the sale.

Demonstration:

```
Secret Word => tango  
Account Number => 123456  
Order Number => 9999999  
Total => 5.99
```

```
md5hash = md5( tango123456999999995.99 )
```

It's important to note that the MD5 hash must also be converted to upper case letters for a clean comparison. How this is done depends on the scripting language that you use. Below are some examples of how to compute the MD5 hash using PHP. This should illustrate how this process works.

The following code would be applicable to orders placed using our Plug and Play cart and our proprietary third party set of parameters.

```
$string_to_hash = "tango123456?" . $_POST["order_number"]  
.  
$_POST["total"];  
$check_key = strtoupper(md5($string_to_hash));  
  
echo ("Returned MD5 Hash : " . $_POST["key"]  
 . "<BR>");  
echo ("Should be : " . $check_key . "<BR>");  
  
if($check_key == $_POST["key"]){  
// At this point the expected key and the returned key match, so the customer  
should be given access to the download  
// This is where you would want to put the code or page for the download  
echo ("<center>They match!</center>"); }  
else {  
// At this point the keys do not match, so either the attempt was fraudulent  
//or a demo order  
// This is where you would put the code or page for an unsuccessful attempt  
echo ("<center>They do NOT match! Was this a demo order?</center>");}
```

The following code would then be applicable to orders placed using the Authorize.net parameter set.

```
$string_to_hash = "tango123456? . $_POST["x_trans_id"]  
.  
$_POST["x_amount"];  
$check_key = strtoupper(md5($string_to_hash));  
  
echo ("Returned MD5 Hash : " . $_POST["x_md5_hash"]  
 . "<BR>");  
echo ("Should be : " . $check_key . "<BR>");
```

```
if($check_key == $_POST["x_md5_hash"]){
// At this point the expected key and the returned key match, so the customer
should be given access to the download
// This is where you would want to put the code or page for the download
echo ("<center>They match!</center>"); }
else {
// At this point the keys do not match, so either the attempt was fraudulent
or a demo order
// This is where you would put the code or page for an unsuccessful attempt echo
("<center>They do NOT match! Was this a demo order?</center>");}
```

INS NOTIFICATIONS

The MD5 hash is also provided to help you verify the authenticity of INS posts. The MD5 hash that is sent with INS posts is a hash of `sale_id+vendor_id+invoice_id+secret word` in the `md5_hash` parameter.

Demonstration:

```
sale_id => 9999999999
vendor_id => 123456
invoice_id => 1111111111
Secret Word => tango
md5hash = md5( 9999999999123456111111111111tango )
```

The following code would be applicable to all INS notifications.

```
$string_to_hash = $_POST["sale_id"] . "123456" . $_POST["invoice_id"] . "tango";
$check_key = strtoupper(md5($string_to_hash));
echo ("Returned MD5 Hash : " . $_POST["md5_hash"]
. "
");
echo ("Should be : " . $check_key . "");
if($check_key == strtoupper($_POST["md5_hash"])){
// If the expected key and the returned key match the authenticity of the message
has been validated.
echo ("They match!"); }
else {
// At this point the keys do not match.
// This is where you would put the code for an unsuccessful attempt.
echo ("They do NOT match!");}
```

Please note that help with implementing the MD5 hash into your return script is beyond the realm of 2Checkout.com's support. This document is provided merely as a reference document to help point you in the right direction. How the MD5 hash is computed is dependent upon the scripting language that you use. Implementation of any MD5 hash checking is solely on your end or your server. 2Checkout.com cannot provide you with support in implementing this or troubleshooting your implementation. We provide you with the hashes as a convenience to help you protect your digital goods.

The following links may be of interest to you if you are looking for more information on the MD5 algorithm and its use.

<http://userpages.umbc.edu/~mabzug1/cs/md5/md5.html>

<http://en.wikipedia.org/wiki/MD5>

<http://msdn.microsoft.com/library/en-us/cpref/html/frlrfSystemSecurityPolicyHashClassMD5Topic.asp>

SALE NOTIFICATIONS

You can use the Instant Notification Service (INS) to automate order management processes by accepting order information via web posts. The INS is a service which will post sets of CGI parameters to any URL you specify. Each post represents a message containing all the information you need about a specific event (such as when a recurring order re-bills successfully). You can enable INS notifications by selecting the **Notifications** tab. From here, you can use the notification interface to enable or disable messages and set either a Global URL for receipt of all messages or individual URLs per message as needed. You may also view and resend Successful and Failed INS notifications by clicking the **Success** and **Failed** sub-categories on the Notifications page. By clicking the **Test** sub-category you may send test INS posts from inside your account to your URL's.

INS notifications are not sent on demo sales so you should use either a live sale or our INS simulator at developers.2checkout.com to test your INS script.

Below is an example PHP class that will check against the MD5 hash returned and provide you with an object to work with.

```
class TCO_INS
{
// Construct variables
public function __construct() // Construct variables
{
$this->sid = "123456"; //2CO account number.
$this->secret_word = "tango"; //2CO secret word
}

public function checkHash() { //Checks MD5 hash and assigns parameters to array.
$insMessage = array();
foreach ($_POST as $k => $v)
{
$insMessage[$k] = $v;
}

$order_number = $insMessage['sale_id'];
$invoice_number = $insMessage['invoice_id'];

$stringToHash = strtoupper(md5($order_number . $this->sid .
$invoice_number . $this->secret_word)); //strtoupper(md5(ordernumber . 2Checkout
Account Number . invoice_id . secret word))

if ($stringToHash != $insMessage['md5_hash']) {
die('Hash Incorrect');
} else {
return $insMessage;
}
}

$checkINS = new TCO_INS(); //Creates new instance.
$insMessage = $checkINS->checkHash(); //Checks MD5 hash and assigns parameters to
array.
echo '<pre>';
print_r($insMessage); //Prints array
echo '</pre>';
```

MESSAGES SUPPORTED & INS MESSAGE PARAMETERS

Please read the following descriptions on each of the INS messages you can subscribe to and each specific parameter to expect as part of each message.

ORDER CREATED

Order creation is a new order placed online by a customer. You may want to automate inventory management, fulfillment or setup of login credentials based on this message. Recurring order creation is a special case of order creation; if any item that is ordered is recurring, the recurring parameter will be set to 1, otherwise, it will be set to 0. PLEASE NOTE: THE INITIAL INSTALLMENT BILLED OFTEN DIFFERS IN PRICE FROM THE INSTALLMENTS DUE TO STARTUP FEES. THIS IS AN INVOICE LEVEL MESSAGE; IT WILL BE SENT ONCE FOR EACH NEW SALE AND WILL CONTAIN INFORMATION ABOUT ALL ITEMS ORDERED.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+ Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

auth_exp

The date credit authorization will expire; format YYYY-MM-DD.

invoice_status

Lower Case Text; Status of a transaction. (approved, pending, deposited, or declined)

fraud_status

Lower Case Text; Status of 2Checkout fraud review (pass, fail, or wait); this parameter could be empty for some sales.

invoice_list_amount
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

invoice_usd_amount
Numeric, Total in US Dollars; format with 2 decimal places.

invoice_cust_amount
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

customer_first_name
Text; Customer's first name. (May not be available on older sales)

customer_last_name
Text; Customer's last name. (May not be available on older sales)

customer_name
Text; Customer's full name. (Name as it appears on credit card)

customer_email
Text; Customer's email address.

customer_phone
Numeric; Customer's phone number; all but digits stripped out.

customer_ip
Text; Customer's IP address at time of sale.

customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)

bill_street_address
Text; Billing street address.

bill_street_address2
Text; Billing street address line 2.

bill_city
Text; Billing address city.

bill_state
Text; Billing address state or province.

bill_postal_code
Text; Billing address postal code.

bill_country
Text; 3-Letter ISO country code of billing address.

ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)

ship_tracking_number
Text; Tracking Number as entered in Admin.

ship_name
Text; Shipping Recipient's name. (As it should appears on shipping label)

ship_street_address
Text; Shipping street address.

ship_street_address2
Text; Shipping street address line 2.

ship_city
Text; Shipping address city.

ship_state
Text; Shipping address state or province.

ship_postal_code
Text; Shipping address postal code.

ship_country
Upper Case Text; 3-Letter ISO country code of shipping address.

item_count
Numeric; Indicates how many numbered sets of item parameters to expect.

item_name_#
Text; Product name.

item_id_#
Text; Product ID.

item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_usd_amount_#
Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#
Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#
Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#
Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#
Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#
Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#
Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#
Numeric; The number of successful recurring installments successfully billed.

FRAUD STATUS CHANGED

All orders must pass 2Checkout's fraud review before they will be billed; therefore we suggest that goods should not be delivered until they have passed this fraud review. If an order fails the fraud review it will also be canceled. An order may change fraud status more than once. This is an invoice level message; it will be sent once for each fraud status change on a sale and will contain information about all items ordered.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

auth_exp

The date credit authorization will expire; format YYYY-MM-DD.

invoice_status

Lower Case Text; Status of a transaction. (approved, pending, deposited, or declined)

fraud_status

Lower Case Text; Status of 2Checkout fraud review (pass, fail, or wait); this parameter could be empty for some sales.

invoice_list_amount

Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

invoice_usd_amount
Numeric, Total in US Dollars; format with 2 decimal places.

invoice_cust_amount
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

customer_first_name
Text; Customer's first name. (May not be available on older sales)

customer_last_name
Text; Customer's last name. (May not be available on older sales)

customer_name
Text; Customer's full name. (Name as it appears on credit card)

customer_email
Text; Customer's email address.

customer_phone
Numeric; Customer's phone number; all but digits stripped out.

customer_ip
Text; Customer's IP address at time of sale.

customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)

bill_street_address
Text; Billing street address.

bill_street_address2
Text; Billing street address line 2.

bill_city
Text; Billing address city.

bill_state
Text; Billing address state or province.

bill_postal_code
Text; Billing address postal code.

bill_country
Text; 3-Letter ISO country code of billing address.

ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)

ship_tracking_number
Text; Tracking Number as entered in Admin.

ship_name
Text; Shipping Recipient's name. (As it should appears on shipping label)

ship_street_address
Text; Shipping street address.

ship_street_address2
Text; Shipping street address line 2.

ship_city
Text; Shipping address city.

ship_state
Text; Shipping address state or province.

ship_postal_code
Text; Shipping address postal code.

ship_country

Upper Case Text; 3-Letter ISO country code of shipping address.

item_count
Numeric; Indicates how many numbered sets of item parameters to expect.

item_name_#
Text; Product name.

item_id_#
Text; Product ID.

item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_usd_amount_#
Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#
Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#
Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#
Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#
Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#
Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#
Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#
Numeric; The number of successful recurring installments successfully billed.

SHIPPING STATUS CHANGED

This message indicates that the ship_status of an order has changed, from not_shipped to shipped (Ex. when an order is marked shipped). This is an invoice level message; it will be sent once for each shipping status change on an invoice and will contain information about all items ordered.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

auth_exp

The date credit authorization will expire; format YYYY-MM-DD.

invoice_status

Lower Case Text; Status of a transaction. (approved, pending, deposited, or declined)

fraud_status

Lower Case Text; Status of 2Checkout fraud review (pass, fail, or wait); this parameter could be empty for some sales.

invoice_list_amount

Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

invoice_usd_amount
Numeric, Total in US Dollars; format with 2 decimal places.

invoice_cust_amount
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

customer_first_name
Text; Customer's first name. (May not be available on older sales)

customer_last_name
Text; Customer's last name. (May not be available on older sales)

customer_name
Text; Customer's full name. (Name as it appears on credit card)

customer_email
Text; Customer's email address.

customer_phone
Numeric; Customer's phone number; all but digits stripped out.

customer_ip
Text; Customer's IP address at time of sale.

customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)

bill_street_address
Text; Billing street address.

bill_street_address2
Text; Billing street address line 2.

bill_city
Text; Billing address city.

bill_state
Text; Billing address state or province.

bill_postal_code
Text; Billing address postal code.

bill_country
Text; 3-Letter ISO country code of billing address.

ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)

ship_tracking_number
Text; Tracking Number as entered in Admin.

ship_name
Text; Shipping Recipient's name. (As it should appears on shipping label)

ship_street_address
Text; Shipping street address.

ship_street_address2
Text; Shipping street address line 2.

ship_city
Text; Shipping address city.

ship_state
Text; Shipping address state or province.

ship_postal_code
Text; Shipping address postal code.

ship_country
Upper Case Text; 3-Letter ISO country code of shipping address.

item_count

Numeric; Indicates how many numbered sets of item parameters to expect.

item_name_#
Text; Product name.

item_id_#
Text; Product ID.

item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_usd_amount_#
Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#
Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#
Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#
Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#
Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#
Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#
Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#
Numeric; The number of successful recurring installments successfully billed.

INVOICE STATUS CHANGED

This status indicates the progress of the payment process. Most orders go from 'approved' to 'pending' to 'deposited', at which point they are applied to the account balance. This is an invoice level message; it will be sent once for each status change on an invoice and will contain information about all items ordered.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

auth_exp

The date credit authorization will expire; format YYYY-MM-DD.

invoice_status

Lower Case Text; Status of a transaction. (approved, pending, deposited, or declined)

fraud_status

Lower Case Text; Status of 2Checkout fraud review (pass, fail, or wait); this parameter could be empty for some sales.

invoice_list_amount

Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

invoice_usd_amount
Numeric, Total in US Dollars; format with 2 decimal places.

invoice_cust_amount
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

customer_first_name
Text; Customer's first name. (May not be available on older sales)

customer_last_name
Text; Customer's last name. (May not be available on older sales)

customer_name
Text; Customer's full name. (Name as it appears on credit card)

customer_email
Text; Customer's email address.

customer_phone
Numeric; Customer's phone number; all but digits stripped out.

customer_ip
Text; Customer's IP address at time of sale.

customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)

bill_street_address
Text; Billing street address.

bill_street_address2
Text; Billing street address line 2.

bill_city
Text; Billing address city.

bill_state
Text; Billing address state or province.

bill_postal_code
Text; Billing address postal code.

bill_country
Text; 3-Letter ISO country code of billing address.

ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)

ship_tracking_number
Text; Tracking Number as entered in Admin.

ship_name
Text; Shipping Recipient's name. (As it should appears on shipping label)

ship_street_address
Text; Shipping street address.

ship_street_address2
Text; Shipping street address line 2.

ship_city
Text; Shipping address city.

ship_state
Text; Shipping address state or province.

ship_postal_code
Text; Shipping address postal code.

ship_country
Upper Case Text; 3-Letter ISO country code of shipping address.

item_count

Numeric; Indicates how many numbered sets of item parameters to expect.

item_name_#
Text; Product name.

item_id_#
Text; Product ID.

item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_usd_amount_#
Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#
Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#
Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#
Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#
Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#
Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#
Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#
Numeric; The number of successful recurring installments successfully billed.

REFUND ISSUED

A refund is any credit issued back to the customer. Refunds can be for specific products / items, or “partial” (no specific product is specified). They will always be specific to a single invoice. This message will allow you to automate more accurate inventory and financial tracking. This is an item level message; it will be sent once for each refund item issued (product or partial) and will only contain information about that item.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

customer_first_name

Text; Customer's first name. (May not be available on older sales)

customer_last_name

Text; Customer's last name. (May not be available on older sales)

customer_name

Text; Customer's full name. (Name as it appears on credit card)

customer_email

Text; Customer's email address.

customer_phone

Numeric; Customer's phone number; all but digits stripped out.

customer_ip
Text; Customer's IP address at time of sale.

customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)

bill_street_address
Text; Billing street address.

bill_street_address2
Text; Billing street address line 2.

bill_city
Text; Billing address city.

bill_state
Text; Billing address state or province.

bill_postal_code
Text; Billing address postal code.

bill_country
Text; 3-Letter ISO country code of billing address.

ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)

ship_tracking_number
Text; Tracking Number as entered in Admin.

ship_name
Text; Shipping Recipient's name. (As it should appears on shipping label)

ship_street_address
Text; Shipping street address.

ship_street_address2
Text; Shipping street address line 2.

ship_city
Text; Shipping address city.

ship_state
Text; Shipping address state or province.

ship_postal_code
Text; Shipping address postal code.

ship_country
Upper Case Text; 3-Letter ISO country code of shipping address.

item_count
Numeric; Indicates how many numbered sets of item parameters to expect.

item_name_#
Text; Product name.

item_id_#
Text; Product ID.

item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_usd_amount_#
Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#

Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#

Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#

Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#

Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#

Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#

Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#

Numeric; The number of successful recurring installments successfully billed.

RECURRING INSTALLMENT SUCCESS

Recurring orders bill automatically according to schedule. This message will allow you to automate any processes which should be keyed off successful installment billing, such as extension of login credentials. This message will send the same customer information and sale_id (and vendor_order_id, if provided) as the original Order Created message, however the successful billing will have generated a new invoice_id and will have incremented the item_rec_install_billed_1 parameter. This is an item level message; it will be sent once for each recurring item billed successfully and will only contain information about that item. PLEASE NOTE: THIS MESSAGE WILL BE IDENTICAL (EXCEPT FOR TIMESTAMP) WHETHER IT IS THE FIRST ATTEMPT OR A RETRY.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

customer_first_name

Text; Customer's first name. (May not be available on older sales)

customer_last_name

Text; Customer's last name. (May not be available on older sales)

customer_name

Text; Customer's full name. (Name as it appears on credit card)

customer_email
Text; Customer's email address.

customer_phone
Numeric; Customer's phone number; all but digits stripped out.

customer_ip
Text; Customer's IP address at time of sale.

customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)

bill_street_address
Text; Billing street address.

bill_street_address2
Text; Billing street address line 2.

bill_city
Text; Billing address city.

bill_state
Text; Billing address state or province.

bill_postal_code
Text; Billing address postal code.

bill_country
Text; 3-Letter ISO country code of billing address.

ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)

ship_tracking_number
Text; Tracking Number as entered in Admin.

ship_name
Text; Shipping Recipient's name. (As it should appear on shipping label)

ship_street_address
Text; Shipping street address.

ship_street_address2
Text; Shipping street address line 2.

ship_city
Text; Shipping address city.

ship_state
Text; Shipping address state or province.

ship_postal_code
Text; Shipping address postal code.

ship_country
Upper Case Text; 3-Letter ISO country code of shipping address.

item_count
Numeric; Indicates how many numbered sets of item parameters to expect.

item_name_#
Text; Product name.

item_id_#
Text; Product ID.

item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_usd_amount_#
Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#

Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#
Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#
Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#
Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#
Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#
Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#
Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#
Numeric; The number of successful recurring installments successfully billed.

RECURRING INSTALLMENT FAILED

Recurring orders bill automatically according to schedule, but billings are not always successful. This message will allow you to automate any processes which should be keyed off installment billing failure, such as suspension of login credentials. This message will send the information based on the last successful installment billed, in other words, the invoice number is the last invoice successfully billed, installments_billed is not incremented, and date_next will often be a date in the past, indicating the original due date for the missed billing. This is an item level message; it will be sent once for each recurring item which fails to bill and will only contain information about that item. PLEASE NOTE: THIS MESSAGE WILL BE IDENTICAL (EXCEPT FOR DATES) WHETHER IT IS THE FIRST ATTEMPT OR A RETRY.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

customer_first_name

Text; Customer's first name. (May not be available on older sales)

customer_last_name

Text; Customer's last name. (May not be available on older sales)

customer_name

Text; Customer's full name. (Name as it appears on credit card)
customer_email
Text; Customer's email address.
customer_phone
Numeric; Customer's phone number; all but digits stripped out.
customer_ip
Text; Customer's IP address at time of sale.
customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)
bill_street_address
Text; Billing street address.
bill_street_address2
Text; Billing street address line 2.
bill_city
Text; Billing address city.
bill_state
Text; Billing address state or province.
bill_postal_code
Text; Billing address postal code.
bill_country
Text; 3-Letter ISO country code of billing address.
ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)
ship_tracking_number
Text; Tracking Number as entered in Admin.
ship_name
Text; Shipping Recipient's name. (As it should appears on shipping label)
ship_street_address
Text; Shipping street address.
ship_street_address2
Text; Shipping street address line 2.
ship_city
Text; Shipping address city.
ship_state
Text; Shipping address state or province.
ship_postal_code
Text; Shipping address postal code.
ship_country
Upper Case Text; 3-Letter ISO country code of shipping address.
item_count
Numeric; Indicates how many numbered sets of item parameters to expect.
item_name_#
Text; Product name.
item_id_#
Text; Product ID.
item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)
item_usd_amount_#
Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#
Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#
Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#
Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#
Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#
Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#
Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#
Numeric; The number of successful recurring installments successfully billed.

RECURRING STOPPED

Recurring orders can be stopped by 2Checkout staff for a variety of reasons, the most common being per customer request. In this most common case, a customer calls 2Checkout and requests to stop an order. For example, John signed up for hosting from you, agreeing to pay \$5/month for a year, however 10 months later, he has no need for his website anymore, and does not want to be billed for the last 2 installments. Because credit card regulations favor the customer in this case, 2Checkout customer care will stop the recurring order, which means that you may want to shut off John's access to your hosting service, in order to free up those resources so they can be used by another customer. This is an item level message; it will be sent once for each recurring item stopped and will only contain information about that item.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

customer_first_name

Text; Customer's first name. (May not be available on older sales)

customer_last_name

Text; Customer's last name. (May not be available on older sales)

customer_name
Text; Customer's full name. (Name as it appears on credit card)

customer_email
Text; Customer's email address.

customer_phone
Numeric; Customer's phone number; all but digits stripped out.

customer_ip
Text; Customer's IP address at time of sale.

customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)

bill_street_address
Text; Billing street address.

bill_street_address2
Text; Billing street address line 2.

bill_city
Text; Billing address city.

bill_state
Text; Billing address state or province.

bill_postal_code
Text; Billing address postal code.

bill_country
Text; 3-Letter ISO country code of billing address.

ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)

ship_tracking_number
Text; Tracking Number as entered in Admin.

ship_name
Text; Shipping Recipient's name. (As it should appears on shipping label)

ship_street_address
Text; Shipping street address.

ship_street_address2
Text; Shipping street address line 2.

ship_city
Text; Shipping address city.

ship_state
Text; Shipping address state or province.

ship_postal_code
Text; Shipping address postal code.

ship_country
Upper Case Text; 3-Letter ISO country code of shipping address.

item_count
Numeric; Indicates how many numbered sets of item parameters to expect.

item_name_#
Text; Product name.

item_id_#
Text; Product ID.

item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_usd_amount_#

Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#
Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#
Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#
Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#
Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#
Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#
Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#
Numeric; The number of successful recurring installments successfully billed.

RECURRING COMPLETE

Recurring orders bill automatically according to schedule, which can end in a fixed amount of time after original order placement if the product's billing duration isn't set to bill "forever". This message can automate any process which should be keyed off completion of installment billing, such as issuing permanent credentials or account closure. This is an item level message; it will be sent once for each recurring item completed and will only contain information about that item.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+ Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

customer_first_name

Text; Customer's first name. (May not be available on older sales)

customer_last_name

Text; Customer's last name. (May not be available on older sales)

customer_name

Text; Customer's full name. (Name as it appears on credit card)

customer_email

Text; Customer's email address.

customer_phone
Numeric; Customer's phone number; all but digits stripped out.

customer_ip
Text; Customer's IP address at time of sale.

customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)

bill_street_address
Text; Billing street address.

bill_street_address2
Text; Billing street address line 2.

bill_city
Text; Billing address city.

bill_state
Text; Billing address state or province.

bill_postal_code
Text; Billing address postal code.

bill_country
Text; 3-Letter ISO country code of billing address.

ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)

ship_tracking_number
Text; Tracking Number as entered in Admin.

ship_name
Text; Shipping Recipient's name. (As it should appears on shipping label)

ship_street_address
Text; Shipping street address.

ship_street_address2
Text; Shipping street address line 2.

ship_city
Text; Shipping address city.

ship_state
Text; Shipping address state or province.

ship_postal_code
Text; Shipping address postal code.

ship_country
Upper Case Text; 3-Letter ISO country code of shipping address.

item_count
Numeric; Indicates how many numbered sets of item parameters to expect.

item_name_#
Text; Product name.

item_id_#
Text; Product ID.

item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_usd_amount_#
Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#
Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#
Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#
Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#
Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#
Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#
Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#
Numeric; The number of successful recurring installments successfully billed.

RECURRING RESTARTED

After being stopped, recurring orders can be restarted by 2Checkout staff for a variety of reasons, the most common being per customer request. In this most common case, a customer calls 2Checkout and requests to restart billing on a recurring order. For example, John signed up for hosting, agreeing to pay \$5/month for a year, however 10 months later, he forgot what the billings he saw on his card were for, called up and had it stopped, and then remembered, and asked to have it restarted. If you turned access off based on the stop message, you will likely also want to turn John's access back on after receiving the restart message. This is an item level message; it will be sent once for each recurring item restarted and will only contain information about that item.

Expected Parameters

message_type

Upper Case Text; Indicates type of message (ORDER_CREATED, FRAUD_STATUS_CHANGED, SHIP_STATUS_CHANGED, INVOICE_STATUS_CHANGED, REFUND_ISSUED, RECURRING_INSTALLMENT_SUCCESS, RECURRING_INSTALLMENT_FAILED, RECURRING_STOPPED, RECURRING_COMPLETE, or RECURRING_RESTARTED)

message_description

Text; Human readable description of message_type.

timestamp

Timestamp of event; format YYYY-MM-DD HH:MM:SS ZZZ.

md5_hash

Upper Case Text; UPPERCASE(MD5_ENCRYPTED(sale_id+vendor_id+invoice_id+Secret Word)); Can be used to validate authenticity of message.

message_id

Numeric; This number is incremented for each message.

key_count

Numeric; Indicates the number of parameters sent in message.

vendor_id

Numeric; 2Checkout account number.

sale_id

Numeric; 2Checkout sale number.

sale_date_placed

Date of sale; format YYYY-MM-DD.

vendor_order_id

Text; Custom order ID, if available. (merchant_order_id parameter can be provided in the CGI hand off to 2Checkout's purchase routine)

invoice_id

Numeric; 2Checkout invoice number; Each sale can have several invoices, most commonly one per installment billed on a recurring order.

recurring

Numeric; returns 1 if any item on the invoice is a recurring item; otherwise it returns 0.

payment_type

Lower Case Text; Customer's payment method. (credit card, online check, paypal ec, OR paypal pay later)

list_currency

Upper Case Text; 3-Letter ISO code for account currency.

cust_currency

Upper Case Text; 3-Letter ISO code for customer currency.

customer_first_name

Text; Customer's first name. (May not be available on older sales)

customer_last_name

Text; Customer's last name. (May not be available on older sales)

customer_name
Text; Customer's full name. (Name as it appears on credit card)

customer_email
Text; Customer's email address.

customer_phone
Numeric; Customer's phone number; all but digits stripped out.

customer_ip
Text; Customer's IP address at time of sale.

customer_ip_country
Text; Country of record for customer's IP address at time of sale; Please note in some cases what is returned is not a country. (Ex. Satellite Provider)

bill_street_address
Text; Billing street address.

bill_street_address2
Text; Billing street address line 2.

bill_city
Text; Billing address city.

bill_state
Text; Billing address state or province.

bill_postal_code
Text; Billing address postal code.

bill_country
Text; 3-Letter ISO country code of billing address.

ship_status
Lower Case Text; Value will be not_shipped, shipped, or empty. (If intangible / does not need shipped)

ship_tracking_number
Text; Tracking Number as entered in Admin.

ship_name
Text; Shipping Recipient's name. (As it should appear on shipping label)

ship_street_address
Text; Shipping street address.

ship_street_address2
Text; Shipping street address line 2.

ship_city
Text; Shipping address city.

ship_state
Text; Shipping address state or province.

ship_postal_code
Text; Shipping address postal code.

ship_country
Upper Case Text; 3-Letter ISO country code of shipping address.

item_count
Numeric; Indicates how many numbered sets of item parameters to expect.

item_name_#
Text; Product name.

item_id_#
Text; Product ID.

item_list_amount_#
Numeric; Total in account pricing currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_usd_amount_#

item_cust_amount_#
Numeric, Total in US Dollars. (format with 2 decimal places)

item_cust_amount_#
Numeric; Total in customer currency; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_type_#
Lower Case Text; Indicates if item is a bill or refund; value will be bill or refund.

item_duration_#
Text; Product duration, how long it re-bills for. (Ex. 1 Year)

item_recurrence_#
Text; Product recurrence, how often it re-bills. (Ex. 1 Month)

item_rec_list_amount_#
Numeric; Product price; format as appropriate to currency. (2 decimal places for most, integer for JPY)

item_rec_status_#
Lower Case Text; indicates status of recurring subscription. (live, canceled, or completed)

item_rec_date_next_#
Date of next recurring installment; format YYYY-MM-DD.

item_rec_install_billed_#
Numeric; The number of successful recurring installments successfully billed.

API

2Checkout's Application Programming Interface (API) provides a way to automate both account and order management processes. Using the API along with the Instant Notification Service (INS) alleviates the necessity of logging into the standard account interface on a day-to-day basis. The API system works by first having you create an API user in your account which has the appropriate permissions to access the API. You are then able to use the supported methods described below to call our API by HTTP POST or GET to update or return sale information at your request. To create a username, please login to your account, click on the **Account** tab and **User Management** sub-category. From here, please click the **Create Username** link and create a new username with API Access and API Updating selected for the Access type. Our API can then be accessed with this new username and password you created.

The API will return XML by default unless the `Accept:` header is passed in the call to define the content type. The API is capable of returning data in the following formats:

- JSON (application/json)
- XML (application/xml) - default
- HTML XOXO microformat (text/html)

Each method we currently support is documented in this section.

Below is an example usage in PHP.

```
$url = "https://www.2checkout.com/api/sales/detail_sale";
$data = array("sale_id" => xxxxxxxxxx);
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_HEADER, "Accept: application/xml");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_USERPWD, "xxxxxxx:xxxxxxx");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
$response = curl_exec($ch);
curl_close($ch);
```

ACCT/DETAIL_COMPANY_INFO

The `detail_company_info` call is used to retrieve your account's company information details from the Site Management page.

URL: `https://www.2checkout.com/api/acct/detail_company_info`

HTTP Method: GET

INPUT PARAMETERS

No input parameters required.

DATA RETURNED

`vendor_id`
Seller ID (Client ID) of user.

`vendor_name`
Company Name that is listed in our purchase routine.

`site_title`
Title that is displayed for you in our product listings.

`site_description`
A brief description of your business and the products or services that you feature.

`soft_descriptor`
Description that is listed on the customer's statement to help identify the charges.

`site_category`
+ Category that your site is listed in.

`return_url`
Url for your customers to be sent to on a successful purchase.

`pending_return_url`
Url for your customers to be sent to when a purchase cannot be verified immediately.

`affiliate_url`
Third party affiliate URL provided by your affiliate program.

`return_method`
Return Method setting. (0 = Given links back to my website, 1 = Direct Return, 2 = Header Redirect)

`secret_word`
Secret word used to calculate the MD5 hash.

`currency_symbol`
Currency Symbol of the currency your account is priced in.

`currency_code`
Currency code of the currency your account is priced in.

`currency_name`
Currency name of the currency your account is priced in.

`url`
URL registered to your account.

`demo`
Account demo setting. (Y = On, N = Off, P = Parameter)

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/acct/detail_company_info \  
-u 'username:password' -H 'Accept: application/json'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "response_code" : "OK",
  "response_message" : "Company information retrieved successfully.",
  "vendor_company_info" : {
    "vendor_id" : 1234567,
    "vendor_name" : "Fake Company",
    "site_title" : "Fake Company",
    "site_description" : "Fake company for testing!",
    "soft_descriptor" : "Test",
    "site_category" : "Electronics",
    "return_url" : "http://www.fake-company.com",
    "pending_return_url" : "http://www.fake-company.com",
    "affiliate_url" : "http://www.fake-company.com",
    "return_method" : "1",
    "secret_word" : "tango",
    "currency_symbol" : "$",
    "currency_code" : "USD",
    "currency_name" : "US Dollars",
    "url" : "www.fake-company.com",
    "demo" : "N"
  }
}
```

ACCT/DETAIL_CONTACT_INFO

The `detail_contact_info` call is used to retrieve your account's contact information details from the Contact Info page.

URL: https://www.2checkout.com/api/acct/detail_contact_info

HTTP Method: GET

INPUT PARAMETERS

No input parameters required.

DATA RETURNED

`mailing_address_id`
Internal ID assigned to your company's mailing address.

`mailing_address_1`
Mailing Address line 1.

`mailing_address_2`
Mailing Address line 2.

`mailing_city`
Mailing Address City.

`mailing_state`
Mailing Address State.

`mailing_postal_code`
Mailing Address Postal Code.

`mailing_country_code`
Mailing Address Country Code.

`physical_address_id`
Internal ID assigned to your company's physical address.

`physical_address_1`
Physical Address line 1.

`physical_address_2`
Physical Address line 2.

`physical_city`
Physical Address City.

`physical_state`
Physical Address State.

`physical_postal_code`
Physical Address Postal Code.

`physical_country_code`
Physical Address Country Code.

`office_phone`
Office Phone Number.

`office_phone_ext`
Business Office Phone Number Extension.

`office_email`
Business Office Email Address.

`customer_service_phone`
Customer Service Phone Number.

`customer_service_phone_ext`:
Customer Service Phone Number Extension.

customer_service_email:
Customer Service Email Address.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/acct/detail_contact_info \  
-H 'Accept: application/json' -u 'username:password'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "response_code" : "OK",  
  "response_message" : "Contact information retrieved successfully.",  
  "vendor_contact_info" : {  
    "mailing_address_id" : "1234567890",  
    "mailing_address_1" : "1785 Obrien Rd",  
    "mailing_address_2" : "",  
    "mailing_city" : "Columbus",  
    "mailing_state" : "Ohio",  
    "mailing_postal_code" : "43228",  
    "mailing_country_code" : "USA",  
    "physical_address_id" : "1234567891",  
    "physical_address_1" : "1785 Obrien Rd",  
    "physical_address_2" : "",  
    "physical_city" : "Columbus",  
    "physical_state" : "Ohio",  
    "physical_postal_code" : "43228",  
    "physical_country_code" : "USA",  
    "office_phone" : "877-294-0273",  
    "office_phone_ext" : "",  
    "office_email" : "noreply@2co.com",  
    "customer_service_phone" : "877-294-0273",  
    "customer_service_phone_ext" : "",  
    "customer_service_email" : "noreply@2co.com"  
  }  
}
```

ACCT/DETAIL_PENDING_PAYMENT

The `detail_pending_payment` call is used to get a detailed estimate of the current pending payment.

URL: https://www.2checkout.com/api/acct/detail_pending_payment

HTTP Method: GET

INPUT PARAMETERS

No input parameters required.

DATA RETURNED

`amount`
Amount of payment.

`reserve_held`
Reserve held.

`reserve_released`
Reserve released.

`total_sales`
Total amount of sales included in payment.

`total_fees`
Total amount of fees included in payment.

`total_refunds`
Total amount of refunds included in payment.

`total_chargeback_fees`
Total amount of chargebacks included in payment.

`total_adjustments`
Total amount of adjustments included in payment.

`total_commissions`
Total amount of commissions earned included in payment.

`total_outgoing_commissions`
Total amount of commissions paid included in payment.

`total_balance_forward`
Total amount of forwarded balance included in payment.

`payment_fee`
Amount charged for sending payment.

`payment_id`
Identifier of payment.

`payment_method`
Method of money transfer.

`release_level`
Release level.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/acct/detail_pending_payment \
-H 'Accept: application/json' -u 'username:password'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "payment" : {
```

```
"amount" : "254.63",
"payment_fee" : "6.00",
"payment_id" : "1234567890",
"payment_method" : "currency wire",
"release_level" : "300",
"reserve_held" : "14.66",
"total_adjustments" : "0.00",
"total_balance_forward" : "0.00",
"total_chargeback_fees" : "0.00",
"total_commissions" : "0.00",
"total_fees" : "17.81",
"total_outgoing_commissions" : "0.00",
"total_refunds" : "0.00",
"total_reserve_released" : "0.00",
"total_sales" : "293.10"
},
"response_code" : "OK",
"response_message" : "payment info retrieved"
}
```

ACCT/LIST_PAYMENTS

The `list_payments` call is used to get a list of past payments.

URL: https://www.2checkout.com/api/acct/list_payments

HTTP Method: GET

INPUT PARAMETERS

No input parameters required.

DATA RETURNED

`payment_id`
Identifier of payment.

`payment_identifier`
Identifier of payment batch.

`date_paid`
Date of payment. (YYYY-MM-DD MM:HH:SS format)

`amount`
Amount of payment.

`reserve_held`
Reserve held.

`total_reserve_released`
Reserve released.

`date_reserve_released`
Date reserve released.

`payment_type`
Method of money transfer.

`total_sales`
Total amount of sales included in payment.

`total_fees`
Total amount of fees included in payment.

`total_refunds`
Total amount of refunds included in payment.

`total_chargeback_fees`
Total amount of chargebacks included in payment.

`total_adjustments`
Total amount of adjustments included in payment.

`total_commissions`
Total amount of commissions earned included in payment.

`total_outgoing_commissions`
Total amount of commissions paid included in payment.

`total_balance_forward`
Total amount of forwarded balance included in payment.

`void`
Whether or not the payment was voided. (0=not voided, 1=voided)

`date_voided`
Date the payment was voided.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/acct/list_payments \
```



```
-H 'Accept: application/json' -u 'username:password'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "payments" : [
    {
      "amount" : "448.04",
      "date_paid" : null,
      "date_reserve_released" : null,
      "date_voided" : null,
      "payment_id" : "1234567890",
      "payment_identifier" : "ach_01012008",
      "payment_type" : "ach",
      "reserve_held" : "25.20",
      "total_adjustments" : "0.00",
      "total_balance_forward" : "0.00",
      "total_chargeback_fees" : "0.00",
      "total_commissions" : "0.00",
      "total_fees" : "30.60",
      "total_outgoing_commissions" : "0.00",
      "total_refunds" : "0.00",
      "total_reserve_released" : "0.00",
      "total_sales" : "503.84",
      "void" : null
    }
  ],
  "response_code" : "OK",
  "response_message" : "payment info retrieved"
}
```

SALES/DETAIL_SALE

The `detail_sale` call is used to retrieve information about a specific sale or invoice.

URL: `https://www.2checkout.com/api/sales/detail_sale`

HTTP Method: GET

INPUT PARAMETERS

`sale_id`

The order number of the requested sale. Optional if `invoice_id` is specified.

`invoice_id`

The invoice number of the requested invoice (specify to include only the requested invoice. Omit and use `sale_id` to include all invoices). Optional if `sale_id` is specified.

DATA RETURNED

`sale`

`sale_id`

Order Number/Sale Number.

`ip_address`

Customer IP Address.

`date_placed`

Date and time the order was placed.

`total`

Sale total amount.

`language`

Language sale was placed in. (values: en,it,fr,no,es_ib,es_la,gr,nl,zh,da,jp,pt,sl,sv,el)

`pay method`

Customer Payment Method.

`cardholder_name`

Customer full name.

`first_six_digits`

First 6 digits of credit card.

`last_two_digits`

Last 2 digits of credit card.

`method`

Method of sale payment. (values: ach, credit card, amex, atmd, bill_me_later, fxsource, paypal_int, paypal_pl, acculynk)

`avs`

AVS match.

`cvv`

CVV match.

`comments`

`username`

User that entered the comment.

`timestamp`

Date and time the comment was entered.

`ip`

IP address of the user.

comment_text
 Comment text.

customer

customer_id
 Customer ID.

title
 Customer Title.

first_name
 Customer first name.

middle_initial
 Customer middle initial.

last_name
 Customer last name.

card_holder_name
 Customer full name.

email
 Customer email address.

phone
 Customer phone number.

phone_extension
 Customer phone extension.

address

address_1
 Billing address line 1.

address_2
 Billing address line 2.

city
 Billing city.

state
 Billing state.

postal_code
 Billing postal code.

country_code
 Billing country code.

country
 Billing country.

invoices

invoice_id
 Invoice ID.

status
 Sale Status. (values: deposited, NULL, approved, declined, pending)

total_usd
 Invoice total in USD.

total_vendor
 Invoice total in account pricing currency.

currency_vendor

Account pricing currency. (values: AED, ARS, AUD, BGN, BRL, CAD, CHF, CLP, DKK, EUR, GBP, HKD, IDR, ILS, INR, JPY, LTL, MXN, MYR, NOK, NZD, PHP, RON, RUB, SEK, SGD, TRY, UAH, USD, ZAR)

total_customer
Invoice total in customer currency.

currency_customer
Customer currency. (values: AED, ARS, AUD, BGN, BRL, CAD, CHF, CLP, DKK, EUR, GBP, HKD, IDR, ILS, INR, JPY, LTL, MXN, MYR, NOK, NZD, PHP, RON, RUB, SEK, SGD, TRY, UAH, USD, ZAR)

fees_2co
2CO transaction fees applied to invoice.

recurring
Recurring. (values: 1=yes, 0=no)

date_placed
Date and time the invoice was placed.

date_shipped
Date and time invoice was shipped.

date_vendor_paid
Date paid.

referrer
Invoice referrer URL.

lineitems

lineitem_id
Line item ID.

vendor_product_id
Seller assigned product ID.

assigned_product_id
2CO assigned product ID.

affiliate_vendor_id
Affiliate ID.

product_description
Product short description.

product_handling
Product handling fee.

product_name
Product Name.

product_price
Line item price.

product_recurrence
Product Recurrence. (values: A number followed by a space followed by "day", "week", "month" or "year")

product_duration
Recurring duration. (values: "Forever" or a number followed by a space followed by "day", "week", "month" or "year")

product_startup_fee
Startup fee.

product_tangible
Tangible or Intangible. (values: 1=intangible, 0=intangible)

product_is_cart
Third Party Cart ID.

installment:
Current installment for recurring billing items.

type:

Line item type. (values: null=products including 2Checkout Signup, signup fee, affiliate, shipping, partial, chargeback, coupon)

options

lineitem_id

Line item ID used with option.

lineitem_option_id

Product option ID.

option_name

Product option name.

option_value

Product option value.

usd_surcharge

surcharge in USD.

vendor_surcharge

Surcharge in pricing currency.

customer_surcharge

Surcharge in customer currency.

billing

bill_id

Billing ID.

amount_usd

Billing amount in USD.

amount_vendor

Billing amount in pricing currency.

amount_customer

Billing amount in customer currency.

installment

Current installment for recurring billing items.

product_recurrence

Billing Recurrence. (values: A number followed by a space followed by "day", "week", "month" or "year")

product_duration

Billing duration. (values: "Forever" or a number followed by a space followed by "day", "week", "month" or "year")

date_next

Next billing date.

date_fail

Last declined date.

date_pending

Billing date pending.

date_deposited

Billing date deposited.

status

Bill Status. (values: bill, refund)

bill_method

Billing method. (values: NULL, ach, credit_card, amex, atmd, bill_me_later, fxsource, paypal_int, paypal_pl, acculynk)

shipping
shipping_id
 Shipment ID for invoice.
date_shipped
 Date and time the invoice was shipped.
tracking_number
 Tracking number entered on invoice.
tracking_url
 Tracking url listed on shipping method.
shipping_method_id
 Shipping method ID.
shipping_method
 Shipping method name.
shipping_name
 Recipient name.
address_1
 Shipping address line 1.
address_2
 Shipping address line 2.
city
 Shipping city.
state
 Shipping state.
postal_code
 Shipping postal code.
country_code
 Shipping country code.
country
 Shipping country.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/sales/detail_sale \  
-H 'Accept: application/json' -d 'invoice_id=1234567890' -u 'username:password'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "response_code" : "OK",  
  "response_message" : "Sale detail retrieved",  
  "sale" : {  
    "comments" : [  
      {  
        "changed_by_ip" : "192.168.0.1",  
        "comment" : "[* Sent: Vendor *] Refund Issued On Sale: test",  
        "timestamp" : "2011-06-11 07:42:05",  
        "username" : "testuser"  
      }  
    ],  
    "customer" : {  
      "address_1" : "1785 Obrien Rd",  
      "address_2" : null,  
      "address_id" : "4473999823",
```

```

"cardholder_name" : "Testing Tester",
"city" : "Columbus",
"country_code" : "USA",
"country_name" : "United States",
"customer_id" : "4473992475",
"email_address" : "no-reply@2co.com",
"first_name" : "Testing",
"lang" : "en",
"last_name" : "Tester",
"middle_initial" : null,
"pay_method" : {
  "avs" : "YY",
  "cvv" : "M",
  "first_six_digits" : "512345",
  "last_two_digits" : "12",
  "method" : "credit card"
},
"phone" : "6149212450",
"phone_ext" : "",
"postal_code" : "43228",
"prefix" : null,
"state" : "OH"
},
"date_placed" : "2011-06-11 07:39:18",
"detail_ip" : {
  "address" : "64.123.456.789",
  "area_code" : 614,
  "city" : "Columbus",
  "country" : "United States",
  "country_code" : "US",
  "region" : "OH",
  "zip" : null
},
"invoices" : [
  {
    "customer_total" : "1.00",
    "date_placed" : "2011-06-11 07:39:18",
    "date_shipped" : "2011-06-11 07:39:18",
    "date_vendor_paid" : null,
    "fees_2co" : "0.51",
    "invoice_id" : "4473999778",
    "lineitems" : [
      {
        "affiliate_vendor_id" : null,
        "billing" : {
          "amount" : "1.00",
          "bill_method" : "credit card",
          "billing_id" : "4473999820",
          "customer_amount" : "1.00",
          "customer_id" : "4473992475",
          "date_deposited" : null,
          "date_end" : "2011-06-18",
          "date_fail" : "1999-01-01",
          "date_next" : null,
          "date_pending" : null,
          "date_start" : "2011-06-13",
          "lineitem_id" : "4473999817",
          "recurring_status" : null,
          "status" : "bill",

```

```

        "usd_amount" : "1.00",
        "vendor_amount" : "1.00"
    },
    "commission" : null,
    "commission_affiliate_vendor_id" : null,
    "commission_flat_rate" : null,
    "commission_percentage" : null,
    "commission_type" : null,
    "commission_usd_amount" : null,
    "customer_amount" : "1.00",
    "flat_rate" : null,
    "installment" : null,
    "invoice_id" : "4473999778",
    "lc_affiliate_vendor_id" : null,
    "lc_usd_amount" : null,
    "lineitem_id" : "4473999817",
    "linked_id" : null,
    "options" : [],
    "percentage" : null,
    "product_description" : "test product",
    "product_duration" : null,
    "product_handling" : "0.00",
    "product_id" : "2267317012",
    "product_is_cart" : "0",
    "product_name" : "download",
    "product_price" : "1.00",
    "product_recurrence" : null,
    "product_startup_fee" : null,
    "product_tangible" : "0",
    "sale_id" : "4473992469",
    "status" : "bill",
    "type" : null,
    "usd_amount" : "1.00",
    "usd_commission" : null,
    "vendor_amount" : "1.00",
    "vendor_product_id" : "Downloads"
},
{
    "affiliate_vendor_id" : null,
    "billing" : {
        "amount" : "1.00",
        "bill_method" : "credit_card",
        "billing_id" : "4474000396",
        "customer_amount" : "1.00",
        "customer_id" : "4473992475",
        "date_deposited" : null,
        "date_end" : null,
        "date_fail" : "1999-01-01",
        "date_next" : null,
        "date_pending" : null,
        "date_start" : "2011-06-11",
        "lineitem_id" : "4474000393",
        "recurring_status" : null,
        "status" : "refund",
        "usd_amount" : "1.00",
        "vendor_amount" : "1.00"
    },
    "commission" : null,
    "commission_affiliate_vendor_id" : null,

```



```

        "commission_flat_rate" : null,
        "commission_percentage" : null,
        "commission_type" : null,
        "commission_usd_amount" : null,
        "customer_amount" : "1.00",
        "flat_rate" : null,
        "installment" : null,
        "invoice_id" : "4473999778",
        "lc_affiliate_vendor_id" : null,
        "lc_usd_amount" : null,
        "lineitem_id" : "4474000393",
        "linked_id" : "4473999817",
        "options" : [],
        "percentage" : null,
        "product_description" : "test product",
        "product_duration" : null,
        "product_handling" : "0.00",
        "product_id" : "2267317012",
        "product_is_cart" : "0",
        "product_name" : "download",
        "product_price" : "1.00",
        "product_recurrence" : null,
        "product_startup_fee" : null,
        "product_tangible" : "0",
        "sale_id" : "1234567890",
        "status" : "refund",
        "type" : null,
        "usd_amount" : "1.00",
        "usd_commission" : null,
        "vendor_amount" : "1.00",
        "vendor_product_id" : "Downloads"
    }
],
"recurring" : "0",
"referrer" : "http://www.acmeonlinebooks.com/",
"sale_id" : "1234568790",
"shipping" : {
    "address_1" : "1785 Obrien Rd",
    "address_2" : null,
    "city" : "Columbus",
    "country_code" : "USA",
    "country_name" : "United States",
    "date_shipped" : "2011-06-11 07:39:18",
    "invoice_id" : "4473999778",
    "postal_code" : "43228",
    "shipping_address_id" : "4473999832",
    "shipping_id" : "4473999814",
    "shipping_method_id" : null,
    "shipping_method_name" : "N/A",
    "shipping_name" : "Testing Tester",
    "state" : "OH",
    "tracking_number" : null,
    "tracking_url" : null
},
"status" : "approved",
"usd_total" : "1.00",
"vendor_id" : "532001",
"vendor_order_id" : null,
"vendor_total" : "1.00"

```

```
    }  
  ],  
  "ip_address" : "64.123.456.789",  
  "ip_country" : "United States",  
  "recurring_decline" : null,  
  "sale_id" : "1234568790"  
}  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to sale.

SALES/LIST_SALES

The `list_sales` call is used to retrieve a summary of all sales or only those matching a variety of sale attributes.

URL: `https://www.2checkout.com/api/sales/list_sales`

HTTP Method: GET

INPUT PARAMETERS

`sale_id`

Search for sale with this Sale ID. Optional.

`invoice_id`

Search for a Sale with this Invoice ID. Optional.

`customer_name`

Search for sales with this cardholder name. Must be at least 3 chars and can be substring of cardholder name. Case Insensitive. Optional.

`customer_email`

Search for sales with this customer email. Can be substring of the email. Case insensitive. Optional.

`customer_phone`

Search for sales with this phone number. Can be an incomplete number but must match from the beginning. Optional.

`vendor_product_id`

Search for sales with this product id. Can be substring of the id. Optional.

`ccard_first6`

Search for sales with these First 6 numbers of the credit card number. Optional.

`ccard_last2`

Search for sales with these Last 2 numbers of the credit card number. Optional.

`sale_date_begin`

Search for sales from this date to current date (or `sale_date_end`). Optional.

`sale_date_end`

Search for sales from beginning of time (or `sale_date_begin`) to this date. Optional.

`declined_recurrings`

Search for declined recurring sales. Optional.

`active_recurrings`

Search for active recurring sales. Optional.

`refunded`

Search for sales that have been refunded in full or partially. Optional.

`cur_page`

The page number to retrieve. First page = 1. Optional.

`pagesize`

Total rows per page. Possible values are 1-100. If pagesize not specified, default of 20 items per page will be assigned internally. Optional.

`sort_col`

The name of the column to sort on. Possible values are `sale_id`, `date_placed`, `customer_name`, `recurring`, `recurring_declined` and `usd_total`. (case insensitive) Optional.

`sort_dir`

The direction of the sort process. ('ASC' or 'DESC') (case insensitive) Optional.

DATA RETURNED

`sale_id`

Order number/Sale Number.
date_placed
Date the order was placed.
customer_name
Customer's name.
recurring
Is the order recurring. (values: 0=No, 1=Yes)
recurring_declined
Date of most recent recurring decline.
USD_total
Customer total in USD.
sale_url
Direct URL to sale detail page.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/sales/list_sales \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'ccard_last2=12' -d 'sort_col=date_placed' -d 'sort_dir=desc'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "page_info" : {  
    "cur_page" : "1",  
    "first_entry" : 1,  
    "first_page" : 1,  
    "first_page_url" : "https://www.2checkout.com/api/sales/list_sales?cur_page=1",  
    "last_entry" : 20,  
    "last_page" : 16,  
    "last_page_url" : "https://www.2checkout.com/api/sales/list_sales?cur_page=16",  
    "next_page" : 2,  
    "next_page_url" : "https://www.2checkout.com/api/sales/list_sales?cur_page=2",  
    "pagesize" : "20",  
    "previous_page" : null,  
    "total_entries" : "309"  
  },  
  "response_code" : "OK",  
  "response_message" : "Sales summaries retrieved successfully.",  
  "sale_summary" : [  
    {  
      "customer_name" : "Manxa J Kit",  
      "date_placed" : "2009-08-18",  
      "recurring" : 0,  
      "recurring_declined" : null,  
      "sale_id" : 1234567890,  
      "sale_url" :  
        "https://www.2checkout.com/api/sales/detail_sale?sale_id=1234567890"  
      "usd_total" : 1.00,  
    },  
    {  
      "customer_name" : "Manxa J Kit",  
      "date_placed" : "2009-08-19",  
      "recurring" : 0,  
      "recurring_declined" : null,  
      "sale_id" : 1234567890,  
    }  
  ]  
}
```

```
"sale_url" :  
"https://www.2checkout.com/api/sales/detail_sale?sale_id=1234567890"  
"usd_total" : 5.00,  
},  
{  
  ...additional sale items  
}  
]  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

SALES/REFUND_INVOICE

The `refund_invoice` call is used to attempt to issue a full or partial refund on an invoice. This call will send the `REFUND_ISSUED` INS message.

URL: `https://www.2checkout.com/api/sales/refund_invoice`

HTTP Method: POST

INPUT PARAMETERS

`sale_id`

Order number/sale ID to issue a refund on. Optional when `invoice_id` is specified, otherwise required.

`invoice_id`

Invoice ID to issue a refund on. Optional when `sale_id` is specified and sale only has 1 invoice.

`amount`

The amount to refund. Only needed when issuing a partial refund. If an amount is not specified, the remaining amount for the invoice is assumed.

`currency`

Currency type of refund amount. Can be 'usd', 'vendor' or 'customer'. Only required if amount is used.

`category`

ID representing the reason the refund was issued. Required. (values: 1-17 from the following list can be used except for 7 as it is for internal use only) 1 = Did not receive order 2 = Did not like item 3 = Item(s) not as described 4 = Fraud 5 = Other 6 = Item not available 7 = Do Not Use (Internal use only) 8 = No response 9 = Recurring last installment 10 = Cancellation 11 = Billed in error 12 = Prohibited product 13 = Service refunded at sellers request 14 = Non delivery 15 = Not as described 16 = Out of stock 17 = Duplicate

`comment`

Message explaining why the refund was issued. Required. May not contain '<' or '>'. (5000 character max)

DATA RETURNED

`response_code`

Tells the user whether or not the operation was successful

`response_message`

Tells the user why the operation was or was not successful

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/sales/refund_invoice \  
-u 'username:password' -d 'sale_id=1234567890' -d 'amount=1.00' \  
-d 'currency=true' -d 'category=13' -H 'Accept: application/json' \  
-d 'comment=Customer deserved a refund.'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "response_code" : "OK",  
  "response_message" : "Refund added to invoice"  
}
```

COMMON ERROR CODES

`PARAMETER_MISSING`

Required parameter missing: <parameter name>
PARAMETER_INVALID
Invalid value for parameter: <parameter name>
RECORD_NOT_FOUND
Unable to find record.
FORBIDDEN
Access denied to sale.

METHOD-SPECIFIC ERROR CODES

FORBIDDEN
Permission denied to set refund category to 7.
FORBIDDEN
Access denied to invoice.
AMBIGUOUS
Ambiguous request. Multiple invoices on sale. invoice_id parameter
required.
TOO_LOW
Amount must be at least 0.01.
NOTHING_TO_DO
Invoice was already refunded.
TOO_HIGH
Amount greater than remaining balance on invoice.
TOO_LATE
Invoice too old to refund. (WILL OCCUR IF SALE IS OVER 180 DAYS)

SALES/REFUND_LINEITEM

The refund_lineitem call is used to attempt to issue a full or partial refund on an invoice. This call will send the REFUND_ISSUED INS message.

URL: https://www.2checkout.com/api/sales/refund_lineitem

HTTP Method: POST

INPUT PARAMETERS

lineitem_id

Line item to issue refund on. Required.

category

ID representing the reason the refund was issued. Required. (values: 1-17 from the following list can be used except for 7 as it is for internal use only) 1 = Did not receive order 2 = Did not like item 3 = Item(s) not as described 4 = Fraud 5 = Other 6 = Item not available 7 = Do Not Use (Internal use only) 8 = No response from seller 9 = Recurring last installment 10 = Cancellation 11 = Billed in error 12 = Prohibited product 13 = Service refunded at sellers request 14 = Non delivery 15 = Not as described 16 = Out of stock 17 = Duplicate

comment

Message explaining why the refund was issued. Required. May not contain '<' or '>'. (5000 character max)

DATA RETURNED

response_code

Tells the user whether or not the operation was successful

response_message

Tells the user why the operation was or was not successful

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/sales/refund_lineitem \  
-u 'username:password' -d 'lineitem_id=1234567890' -d 'category=16' \  
-d 'comment=Item not in stock.' -H 'Accept: application/json'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "response_code" : "OK",  
  "response_message" : "Lineitem refunded successfully"  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Required parameter is invalid: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to sale.

METHOD-SPECIFIC ERROR CODES

FORBIDDEN

Permission denied to set refund category to 7.

INVALID_PARAMETER

This lineitem cannot be refunded.

NOTHING_TO_DO

Lineitem was already refunded.

TOO_LATE

Invoice too old to refund lineitem. (WILL OCCUR IF SALE IS OVER 180 DAYS)

TOO_HIGH

Lineitem amount greater than remaining balance on invoice.

TOO_LOW

Lineitem amount must be at least 0.01.

SALES/STOP_LINEITEM_RECURRING

The `stop_lineitem_recurring` call is used to attempt to stop a recurring line item for a specified sale. This call will send the `RECURRING_STOPPED` INS message.

URL: `https://www.2checkout.com/api/sales/stop_lineitem_recurring`

HTTP Method: POST

INPUT PARAMETERS

`lineitem_id`

Line Item ID to stop recurring on. Required.

DATA RETURNED

`response_code`

Tells the user whether or not the operation was successful.

`response_message`

Tells the user why the operation was or was not successful.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/sales/stop_lineitem_recurring -u \  
'username:password' -d 'vendor_id=123456' -d 'lineitem_id=1234567890' \  
-H 'Accept: application/json'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "response_code" : "OK",  
  "response_message" : "Recurring billing stopped for lineitem"  
}
```

COMMON ERROR CODES

`PARAMETER_MISSING`

Required parameter missing: <parameter name>

`PARAMETER_INVALID`

Invalid value for parameter: <parameter name>

`RECORD_NOT_FOUND`

Unable to find record.

`FORBIDDEN`

Access denied to sale.

SALES/MARK_SHIPPED

The `mark_shipped` call is used to attempt to mark an order as shipped and will attempt to reauthorize sale if specified in call. This call will send the `SHIP_STATUS_CHANGED` INS message.

URL: `https://www.2checkout.com/api/sales/mark_shipped`

HTTP Method: POST

INPUT PARAMETERS

`sale_id`

The order number/sale ID to mark shipped. Optional when `invoice_id` is present.

`invoice_id`

ID of the invoice to add tracking information to. Required on sales with more than one invoice.

`tracking_number`

The tracking number issued by the shipper. Required.

`cc_customer`

Specify whether the customer should be automatically notified. Defaults to false. Optional.

`reauthorize`

Reauthorize payment if payment authorization has expired. Defaults to false. Optional.

`comment`

Any text except for '`<`' and '`>`' up to 255 chars in length. Optional.

DATA RETURNED

`response_code`

Tells the user whether or not the operation was successful.

`response_message`

Tells the user why the operation was or was not successful.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/sales/mark_shipped \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'tracking_number=123' -d 'cc_customer=true' -d 'sale_id=1234567890'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "response_code" : "OK",  
  "response_message" : "Sale marked shipped."  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: `<parameter name>`

PARAMETER_INVALID

Invalid value for parameter: `<parameter name>`

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to sale.

METHOD SPECIFIC ERROR CODES

NOTHING_TO_DO

Item not shippable.

TOO_LATE

Payment is already pending or deposited and cannot be reauthorized.

TOO_SOON

Please wait until the next day before trying to reauthorize again.

FAILED

Failed to reauthorize payment.

INTERNAL_ERROR

Failed to marked shipped but reauthorization succeeded.

INTERNAL_ERROR

Failed to marked shipped.

SALES/CREATE_COMMENT

The `create_comment` call is used to add a comment to a specified sale.

URL: `https://www.2checkout.com/api/sales/create_comment`

HTTP Method: POST

INPUT PARAMETERS

`sale_id`

The order number/sale ID of a sale to look for. Required.

`sale_comment`

String value of comment to be submitted. Required.

`cc_vendor`

Set to 1 to have a copy sent to seller. Optional.

`cc_customer`

Set to 1 to have the customer sent an email copy. Optional.

DATA RETURNED

`response_code`

Tells the user whether or not the operation was successful.

`response_message`

Tells the user why the operation was or was not successful.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/sales/create_comment -u \  
'username:password' -H 'Accept: application/json' -d 'sale_id=1234567890' \  
-d 'sale_comment=Your order will be shipped this week.'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "response_code" : "OK",  
  "response_message" : "Created comment successfully."  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to sale.

PRODUCTS/DETAIL_PRODUCT

The `detail_product` call is used to retrieve the details for a single product.

URL: `https://www.2checkout.com/api/products/detail_product`

HTTP Method: GET

INPUT PARAMETERS

`product_id`
ID of product to retrieve details for. Required.

DATA RETURNED

`product`
`product_id`
System Product ID.
`name`
Product Name
`vendor_product_id`
Seller Assigned Product ID.
`description`
Product Short Description.
`long_description`
Product Long Description.
`price`
Product Price.
`tangible`
Tangible=1, Intangible=0.
`weight`
Product Weight.
`handling`
Product Handling Fee.
`recurrence`
Product Recurrence.
`startup_fee`
Recurring Startup Fee.
`duration`
Product Duration.
`assigned_product_id`
2CO Assigned Product ID.
`pending_url`
Product Pending URL.
`approved_url`
Product Approved URL.
`commission`
Affiliate commission.
`commission_type`
Affiliate commission type.

`image`
`image_number`

Image number.
image_id
Image ID.

option
option_value_id
Product option value.
option_id
Product option ID.
option_name
Product option name.
vendor_id
Seller ID.
option_value_surcharge
Option value surcharge.
option_value_name
Option value name.

category
category_id
Category ID
name
Category name.
parent_id
Category parent ID.
description
Category description.
parent_name
Category parent name.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/products/detail_product \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'product_id=1'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "product" : {  
    "approved_url" : "http://www.acmeonlinebooks.com/download.php",  
    "assigned_product_id" : "1",  
    "categories" : [  
      {  
        "category_id" : "5",  
        "description" : null,  
        "name" : "Software",  
        "parent_id" : null,  
        "parent_name" : null  
      }  
    ],  
    "commission" : 0,  
    "commission_amount" : null,  
  }  
}
```

```

    "commission_type" : null,
    "description" : "test product",
    "duration" : null,
    "handling" : null,
    "images" : [
      {
        "image_id" : "12454841",
        "image_number" : "1"
      }
    ],
    "long_description" : "Test long Description",
    "name" : "download",
    "options" : [
      {
        "option_id" : "3997272579",
        "option_name" : "test",
        "option_values" : [
          {
            "option_value_id" : "3997272582",
            "option_value_name" : "1",
            "option_value_surcharge" : "0.00"
          },
          {
            "option_value_id" : "3997272585",
            "option_value_name" : "2",
            "option_value_surcharge" : "0.00"
          },
          {
            "option_value_id" : "3997272588",
            "option_value_name" : "3",
            "option_value_surcharge" : "0.00"
          },
          {
            "option_value_id" : "3997272591",
            "option_value_name" : "4",
            "option_value_surcharge" : "0.00"
          }
        ]
      }
    ],
    "pending_url" : null,
    "price" : "1.00",
    "product_id" : "2267317012",
    "recurrence" : null,
    "recurrence_p" : null,
    "recurring" : "0",
    "startup_fee" : null,
    "tangible" : "0",
    "vendor_id" : "532001",
    "vendor_product_id" : "Downloads",
    "weight" : null
  },
  "response_code" : "OK",
  "response_message" : "Product detail information retrieved successfully"
}

```


PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to product.

PRODUCTS/LIST_PRODUCTS

The `list_products` call is used to retrieve list of all products in account.

URL: https://www.2checkout.com/api/products/list_products

HTTP Method: GET

INPUT PARAMETERS

`assigned_product_id`

Filter list results on 2CO ID. Optional.

`vendor_product_id`

Filter list results on product ID. Optional.

`name`

Filter list results on product name. Optional.

`cur_page`

The page number to retrieve. First page = 1. Optional.

`pagesize`

Total rows per page. Possible values are 1-100. If pagesize not specified, default of 20 items per page will be assigned internally. Optional.

`sort_col`

The name of the column to sort on. Possible values are `product_id`, `name`, `price`, `vendor_product_id`, `assigned_product_id`, `tangible`, `weight`, `handling`, `description`, `long_description`, `pending_url`, `approved_url`, `startup_fee`, `recurrence`, `duration`, `description`, `category_id`, `commission_amount`, `option_id` and `commission_type`. (case insensitive) Optional.

`sort_dir`

The direction of the sort process. ('ASC' or 'DESC') (case insensitive) Optional.

DATA RETURNED

`page_info`

`first_page`

Link to the first product page.

`previous_page`

Link to the previous product page.

`next_page`

Link to the next product page.

`last_page`

Link to the last product page.

`first`

First product ID in page.

`last`

Last product ID in page.

`total_entries`

Total number of products.

`current_page`

Current product page.

`pages_in_set`

Number of pages in set.

`entries_per_page`

Product of product entries in set.

products
product_id
 System Product ID.
name
 Product Name.
vendor_product_id
 Seller Assigned Product ID.
description
 Product Short Description.
long_description
 Product Long Description.
price
 Product Price.
tangible
 Tangible=1, Intangible=0
weight
 Product Weight.
handling
 Product Handling Fee.
recurrence
 Product Recurrence.
startup_fee
 Recurring Startup Fee.
duration
 Product Duration.
assigned_product_id
 2CO Assigned Product ID.
pending_url
 Product Pending URL.
approved_url
 Product Approved URL.
commission
 Affiliate Commission.
commission_type
 Affiliate Commission Type.

images

image_number
 Image Number.
image_id
 Image ID.

options

option_value_id
 Option Value ID.
option_id
 Option ID.
option_name
 Option Name.
vendor_id

Seller ID.
option_value_surcharge
Option Value Surcharge.
option_value_name
Option Value Name.

categories
category_id
Category ID.
name
Category Name.
parent_id
Category Parent ID.
description
Category Description.
parent_name
Category Parent Name.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/products/list_products \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'sort_col=product_id' -d 'sort_dir=desc'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "page_info" : {  
    "cur_page" : "1",  
    "first_entry" : 1,  
    "first_page" : 1,  
    "first_page_url" :  
"https://www.2checkout.com/api/products/list_products?cur_page=1",  
    "last_entry" : 20,  
    "last_page" : 12,  
    "last_page_url" :  
"https://www.2checkout.com/api/products/list_products?page=12",  
    "next_page" : 2,  
    "next_page_url" :  
"https://www.2checkout.com/api/products/list_products?cur_page=2",  
    "pagesize" : "20",  
    "previous_page" : null,  
    "total_entries" : "231"  
  },  
  "products" : [  
    {  
      "approved_url" : "http://www.acmeonlinebooks.com/download.php",  
      "assigned_product_id" : "1",  
      "categories" : [  
        {  
          "category_id" : "5",  
          "description" : null,  
          "name" : "Software",  
          "parent_id" : null,  
          "parent_name" : null
```

```

    }
  ],
  "commission" : null,
  "commission_type" : null,
  "description" : "test product",
  "duration" : null,
  "handling" : "0.00",
  "images" : [
    {
      "image_id" : "12454841",
      "image_number" : "1"
    }
  ],
  "long_description" : "Test long description",
  "name" : "download",
  "options" : [
    {
      "option_id" : "3997272579",
      "option_name" : "test",
      "option_values" : [
        {
          "option_value_id" : "3997272582",
          "option_value_name" : "1",
          "option_value_surcharge" : "0.00"
        },
        {
          "option_value_id" : "3997272585",
          "option_value_name" : "2",
          "option_value_surcharge" : "0.00"
        },
        {
          "option_value_id" : "3997272588",
          "option_value_name" : "3",
          "option_value_surcharge" : "0.00"
        },
        {
          "option_value_id" : "3997272591",
          "option_value_name" : "4",
          "option_value_surcharge" : "0.00"
        }
      ]
    }
  ],
  "pending_url" : null,
  "price" : "1.00",
  "product_id" : "2267317012",
  "recurrence" : null,
  "startup_fee" : null,
  "tangible" : "0",
  "vendor_id" : "532001",
  "vendor_product_id" : "Downloads",
  "weight" : "0.00"
},
    {
      ...additional product listings
    }
],
"response_code" : "OK",

```

```
"response_message" : "Product list successfully retrieved."  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

PRODUCTS/CREATE_PRODUCT

The `create_product` call is used to create a new product.

URL: `https://www.2checkout.com/api/products/create_product`

HTTP Method: POST

INPUT PARAMETERS

`name`

Product Name. Required.

`price`

Product Price. Required.

`vendor_product_id`

Seller Assigned Product ID. Optional.

`description`

Product Short Description. Optional.

`long_description`

Product Long Description. Optional.

`pending_url`

Product Pending URL. Optional.

`approved_url`

Product Approved URL. Optional.

`tangible`

Tangible=1, Intangible=0. Optional.

`weight`

Decimal value of weight. Required for tangible products.

`handling`

Specifies handling charge if applicable. Required for tangible products.

`recurring`

1 = recurring, 0 = non-recurring. Required for recurring products.

`startup_fee`

Specifies start up fee if applicable. Optional.

`recurrence`

Specifies recurrence frequency (n Week|Month|Year), if applicable. Required for recurring products.

`duration`

Specifies recurrence duration (n Week|Month|Year|Forever), if applicable. Required for recurring products.

`commission`

1 = commission, 0 = no commission. Required for affiliate products.

`commission_type`

Sets commission type for product. Possible values: amount, percentage. Required for affiliate products.

`commission_amount`

Sets commission value for product (based on `commission_type`). Required for affiliate products.

`option_id`

Accepts single or multiple option IDs to assign to product. (multiple IDs would be `&option_id=xxxxxxx` repeated for each ID to assign, where `xxxxxxx` represents each ID value to assign.) Optional.

`category_id`

Accepts single or multiple category IDs to assign product to. (multiple IDs would be `&category_id=xxxxxxx` repeated for each ID to assign, where `xxxxxxx` represents each ID value to assign.) Optional.

DATA RETURNED

`response_code`
Tells the user whether or not the operation was successful.

`response_message`
Tells the user why the operation was or was not successful.

`assigned_product_id`
ID assigned to the product by 2Checkout.

`product_id`
ID assigned to the product by 2Checkout.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/products/create_product \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'name=test product' -d 'price=1.00' -d 'vendor_product_id=123456789'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "assigned_product_id" : "1290",  
  "product_id" : "4337692361",  
  "response_code" : "OK",  
  "response_message" : "Product successfully created"  
}
```

COMMON ERROR CODES

`PARAMETER_MISSING`
Required parameter missing: <parameter name>

`PARAMETER_INVALID`
Invalid value for parameter: <parameter name>

`RECORD_NOT_FOUND`
Unable to find record.

PRODUCTS/UPDATE_PRODUCT

The `update_product` call is used to update a product.

URL: `https://www.2checkout.com/api/products/update_product`

HTTP Method: POST

INPUT PARAMETERS

`product_id`
ID of product to update. Required.

`name`
Product Name. Required.

`price`
Product Price. Required.

`vendor_product_id`
Seller Assigned Product ID. Optional.

`description`
Product Short Description. Optional.

`long_description`
Product Long Description. Optional.

`pending_url`
Product Pending URL. Optional.

`approved_url`
Product Approved URL. Optional.

`tangible`
Tangible=1, Intangible=0. Optional.

`weight`
Decimal value of weight. Required for tangible products.

`handling`
Specifies handling charge if applicable. Required for tangible products.

`recurring`
1 = recurring, 0 = non-recurring. Required for recurring products.

`startup_fee`
Specifies start up fee if applicable. Optional.

`recurrence`
Specifies recurrence frequency (n Week|Month|Year), if applicable. Required for recurring products.

`duration`
Specifies recurrence duration (n Week|Month|Year|Forever), if applicable. Required for recurring products.

`commission`
1 = commission, 0 = no commission. Required for affiliate products.

`commission_type`
Sets commission type for product. Possible values: amount, percentage. Required for affiliate products.

`commission_amount`
Sets commission value for product (based on `commission_type`). Required for affiliate products.

`option_id`
Accepts single or multiple option IDs to assign to product. (multiple IDs would be `&option_id=xxxxxxx` repeated for each ID to assign, where xxxxxxxx represents each ID value to assign.) Optional.

`category_id`

Accepts single or multiple category IDs to assign product to. (multiple IDs would be &category_id=xxxxxxx repeated for each ID to assign, where xxxxxxxx represents each ID value to assign.) Optional.

DATA RETURNED

response_code

Tells the user whether or not the operation was successful.

response_message

Tells the user why the operation was or was not successful.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/products/update_product \
-H 'Accept: application/json' -u 'username:password' \
-d 'name=test product' -d 'price=1.00' -d 'product_id=1'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "assigned_product_id" : "1",
  "product_id" : "1234567890",
  "response_code" : "OK",
  "response_message" : "Product successfully updated"
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to product.

PRODUCTS/DELETE_PRODUCT

The `delete_product` call is used to delete a product.

URL: `https://www.2checkout.com/api/products/delete_product`

HTTP Method: POST

INPUT PARAMETERS

`product_id`

2CO system assigned product ID to delete. Required.

DATA RETURNED

`response_code`

Tells the user whether or not the operation was successful.

`response_message`

Tells the user why the operation was or was not successful.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/products/delete_product \
-H 'Accept: application/json' -u 'username:password' \
-d 'product_id=1'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "response_code" : "OK",
  "response_message" : "Product successfully deleted.",
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to product.

PRODUCTS/DETAIL_OPTION

The detail_option call is used to retrieve the details for a single option.

URL: https://www.2checkout.com/api/products/detail_option

HTTP Method: GET

INPUT PARAMETERS

option_id

ID value of option to get details for. Required.

DATA RETURNED

option_value_id

System Option ID.

option_value_surcharge

Option value surcharge amount.

option_value_name

Option value name.

option_id

System Option ID.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/products/detail_option \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'option_id=1'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "option" : [  
    {  
      "option_id" : "3997272579",  
      "option_name" : "test",  
      "option_values" : [  
        {  
          "option_value_id" : "3997272582",  
          "option_value_name" : "1",  
          "option_value_surcharge" : "0.00"  
        },  
        {  
          "option_value_id" : "3997272585",  
          "option_value_name" : "2",  
          "option_value_surcharge" : "0.00"  
        },  
        {  
          "option_value_id" : "3997272588",  
          "option_value_name" : "3",  
          "option_value_surcharge" : "0.00"  
        },  
        {  
          "option_value_id" : "3997272591",  
          "option_value_name" : "4",  
          "option_value_surcharge" : "0.00"  
        }  
      ]  
    }  
  ]  
}
```

```
        }
    ]
}
],
"response_code" : "OK",
"response_message" : "Option detail retrieved successfully."
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to option.

PRODUCTS/LIST_OPTIONS

The `list_options` call is used to retrieve list of all options in account.

URL: `https://www.2checkout.com/api/products/list_options`

HTTP Method: GET

INPUT PARAMETERS

`option_name`

Filter list results by option name. Optional.

`option_value_name`

Filter list results by option value name. Optional.

`cur_page`

The page number to retrieve. First page = 1. Optional.

`pagesize`

Total rows per page. Possible values are 1-100. If pagesize not specified, default of 20 items per page will be assigned internally. Optional.

`sort_col`

The name of the column to sort on. Possible values are `option_id` and `option_name`. (case insensitive) Optional.

`sort_dir`

The direction of the sort process. ('ASC' or 'DESC') (case insensitive) Optional.

DATA RETURNED

`options`

`option_id`

System Option ID.

`option_name`

Option name.

`option_values`

`option_id`

System Option ID.

`option_value_id`

System Option ID.

`vendor_id`

Seller ID (Client ID) of user.

`option_value_surcharge`

Option value surcharge.

`option_value_name`

Option value name.

`option_name`

Option name.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/products/list_options \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'sort_col=option_name' -d 'sort_dir=desc'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "options" : [
    {
      "option_id" : "4276589689",
      "option_name" : "size",
      "option_values" : [
        {
          "option_value_id" : "4276589692",
          "option_value_name" : "sds",
          "option_value_surcharge" : "0.00"
        }
      ]
    },
    {
      "option_id" : "4137422637",
      "option_name" : "Size",
      "option_values" : [
        {
          "option_value_id" : "4137422640",
          "option_value_name" : "S",
          "option_value_surcharge" : "1.00"
        },
        {
          "option_value_id" : "4137422643",
          "option_value_name" : "M",
          "option_value_surcharge" : "0.00"
        },
        {
          "option_value_id" : "4137422646",
          "option_value_name" : "L",
          "option_value_surcharge" : "0.00"
        }
      ]
    },
    {
      "option_id" : "4127826491",
      "option_name" : "test",
      "option_values" : [
        {
          "option_value_id" : "4127826494",
          "option_value_name" : "test",
          "option_value_surcharge" : "0.00"
        }
      ]
    }
  ],
  "page_info" : {
    "cur_page" : "1",
    "first_entry" : 1,
    "first_page" : 1,
    "first_page_url" :
      "https://www.2checkout.com/api/products/list_options?cur_page=1",
    "last_entry" : "5",
    "last_page" : 1,
    "last_page_url" :
      "https://www.2checkout.com/api/products/list_options?cur_page=1",
    "next_page" : null,
  }
}
```

```
    "pagesize" : "20",
    "previous_page" : null,
    "total_entries" : "5"
  },
  "response_code" : "OK",
  "response_message" : "Option information retrieved successfully."
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

PRODUCTS/CREATE_OPTION

The `create_option` call is used to create a new product option.

URL: `https://www.2checkout.com/api/products/create_option`

HTTP Method: POST

INPUT PARAMETERS

`option_name`

Name of new product option. Required.

`option_value_name`

Name of option value. Multiples allowed but must equal number of `option_value_surcharge`s supplied (`&option_value_name=xxxxxxx&option_value_name=yyyyyyyyy`). Required.

`option_value_surcharge`

Amount of option value. Multiples allowed but must equal number of `option_value_name`s supplied (`&option_value_surcharge=xxxxxxx&option_value_surcharge=yyyyyyyyy`). Required.

DATA RETURNED

`response_code`

Tells the user whether or not the operation was successful.

`response_message`

Tells the user why the operation was or was not successful.

`option_id`

ID assigned to the option by 2Checkout.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/products/create_option \
-H 'Accept: application/json' -u 'username:password' \
-d 'option_name=test' -d 'option_value_name=red' -d 'option_value_surcharge=1.00' \
-d 'option_value_name=blue' -d 'option_value_surcharge=1.00'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "response_code" : "OK",
  "response_message" : "Option created successfully.",
  "option_id" : "123456789"
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

PRODUCTS/UPDATE_OPTION

The `update_option` call is used to update an option.

URL: `https://www.2checkout.com/api/products/update_option`

HTTP Method: POST

INPUT PARAMETERS

`vendor_id`
Seller ID (Client ID) of user. Optional.

`option_id`
ID of option being updated. Required.

`option_name`
Name of new product option. Required.

`option_value_id`
ID of existing option value to update/delete. Required when updating/deleting an option value.

`option_value_name`
Name of option value. Required when creating an option value.

`option_value_surcharge`
Amount of option surcharge. Required when creating an option value surcharge.

DATA RETURNED

`response_code`
Tells the user whether or not the operation was successful.

`response_message`
Tells the user why the operation was or was not successful.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/products/update_option \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'option_id=1001' -d 'option_name=test option' -d 'option_value_name=colors'  
-d 'option_value_surcharge=1.00'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "response_code" : "OK",  
  "response_message" : "Option updated successfully."  
}
```

COMMON ERROR CODES

`PARAMETER_MISSING`
Required parameter missing: <parameter name>

`PARAMETER_INVALID`
Invalid value for parameter: <parameter name>

`RECORD_NOT_FOUND`
Unable to find record.

`FORBIDDEN`
Access denied to sale.

PRODUCTS/DELETE_OPTION

The delete_option call is used to delete a product option.

URL: https://www.2checkout.com/api/products/delete_option

HTTP Method: POST

INPUT PARAMETERS

option_id

ID of option to delete. Required.

DATA RETURNED

response_code

Tells the user whether or not the operation was successful.

response_message

Tells the user why the operation was or was not successful.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/products/delete_option \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'option_id=1001'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "response_code" : "OK",  
  "response_message" : "Option deleted successfully.",  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to option.

PRODUCTS/DETAIL_COUPON

The `detail_coupon` call is used to retrieve the details for a single coupon.

URL: https://www.2checkout.com/api/products/detail_coupon

HTTP Method: GET

INPUT PARAMETERS

`coupon_code`

The string value of coupon code. Required.

DATA RETURNED

`coupon_id`

System coupon ID.

`coupon_code`

The string value of coupon code.

`date_expire`

Expiration Date. (YYYY-MM-DD)

`percentage_off`

Percentage Discount.

`type`

Coupon type.

`value_off`

Fixed Discount.

`minimum_purchase`

Minimum purchase amount.

`product_id`

ID of product selected for coupon.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/products/detail_coupon \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'coupon_code=discount'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "coupon" : {  
    "coupon_code" : "TESTAPI321",  
    "date_expire" : "2012-06-06",  
    "minimum_purchase" : "1.00",  
    "percentage_off" : "1",  
    "product" : [  
      {  
        "product_id" : "0",  
        "product_url" :  
"https://www.2checkout.com/api/products/detail_product?product_id=0"  
      }  
    ],  
    "type" : "sale",  
    "value_off" : null  
  },  
}
```

```
"response_code" : "OK",  
"response_message" : "Coupon detail retrieved successfully."  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to coupon.

PRODUCTS/LIST_COUPONS

The `list_coupons` call is used to retrieve list of all coupons in the account.

URL: https://www.2checkout.com/api/products/list_coupons

HTTP Method: GET

INPUT PARAMETERS

No input parameters required.

DATA RETURNED

`coupons`

`coupon_code`

The string value of coupon code.

`vendor_id`

Seller ID (Client ID) of user.

`date_expire`

Expiration Date.(YYYY-MM-DD)

`percentage_off`

Percentage Discount.

`value_off`

Fixed Discount.

`type`

Coupon type.

`minimum_purchase`

Minimum purchase amount.

`product_id`

ID of product selected for coupon.

EXAMPLE API CALL USING CURL

```
curl -G https://www.2checkout.com/api/products/list_coupons \
-H 'Accept: application/json' -u 'username:password' \
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "coupon" : [
    {
      "coupon_code" : "Discount",
      "date_expire" : "2011-08-31",
      "minimum_purchase" : "0.01",
      "percentage_off" : "0.10",
      "type" : "product",
      "value_off" : null
    }
  ],
  "response_code" : "OK",
  "response_message" : "Coupon information retrieved successfully."
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

PRODUCTS/CREATE_COUPON

The `create_coupon` call is used to create a new coupon.

URL: `https://www.2checkout.com/api/products/create_coupon`

HTTP Method: POST

INPUT PARAMETERS

`coupon_code`

The string value of coupon code. Optional.

`date_expire`

Expiration date of new coupon. (YYYY-MM-DD) Required.

`type`

Denotes if coupon applies to shipping, sale or product. Required.

`percentage_off`

Percentage, if supplied, to discount from purchase price. Can only be used with sale or product type coupons. (value_off must be NULL, if percentage_off used)

`value_off`

Decimal value of amount to discount. Can only be used with sale or product type coupons. (percentage_off must be NULL, if value_off used)

`minimum_purchase`

Decimal value indicating minimum required purchase amount before discount applies. Optional.

`product_id`

2CO system product ID. Can accept multiple product_ids in query string. (i.e. `product_id=112345678&product_id=987564321`) Required for product coupons if `select_all` is not specified.

`select_all`

If set to true (1), will select all products and override any product_ids supplied. Set to 0 to remove all product assignments. Required for product coupons if `product_id` is not specified.

DATA RETURNED

`response_code`

Tells the user whether or not the operation was successful.

`response_message`

Tells the user why the operation was or was not successful.

`coupon_code`

ID assigned to the product by 2Checkout.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/products/create_coupon \
-H 'Accept: application/json' -u 'username:password' \
-d 'coupon_code=discount' -d 'date_expire=2099-12-22' -d 'type=sale' \
-d 'value_off=5.00' -d 'minimum_purchase=10.00'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "coupon_code" : "TESTAPI321",
  "response_code" : "OK",
```



```
"response_message" : "Coupon successfully created"  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

PRODUCTS/UPDATE_COUPON

The `update_coupon` call is used to update a coupon.

URL: `https://www.2checkout.com/api/products/update_coupon`

HTTP Method: POST

INPUT PARAMETERS

`coupon_code`

The string value of the existing coupon code. Required.

`new_code`

String value of new coupon code. Optional.

`date_expire`

New expiration date of coupon. (YYYY-MM-DD) Optional.

`type`

Denotes if coupon applies to shipping, sale or product. Required.

`percentage_off`

Percentage, if supplied, to discount from purchase price. (value_off must be NULL, if percentage_off used) Optional.

`value_off`

Decimal value of amount to discount. (percentage_off must be NULL, if value_off used) Optional.

`minimum_purchase`

Decimal value indicating minimum purchase required before discount applies. Required if percentage_off or value_off are specified.

`product_id`

Can accept multiple product_ids in querystring. (i.e. `product_id=1112223333&product_id=2223334444`) Required for product coupons if select_all is not specified.

`select_all`

Set to 1 to enable. Set to 0 to remove all product assignments. Required for product coupons if product_id is not specified.

DATA RETURNED

`response_code`

Tells the user whether or not the operation was successful.

`response_message`

Tells the user why the operation was or was not successful.

EXAMPLE API CALL USING CURL

```
curl -X POST https://www.2checkout.com/api/products/update_coupon \  
-H 'Accept: application/json' -u 'username:password' \  
-d 'coupon_code=discount' -d 'date_expire=2099-12-22' -d 'type=sale'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{  
  "coupon_code" : "discount",  
  "response_code" : "OK",  
  "response_message" : "Coupon updated successfully"  
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

PRODUCTS/DELETE_COUPON

The delete_coupon call is used to delete a coupon.

URL: `https://www.2checkout.com/api/products/delete_coupon`

HTTP Method: POST

INPUT PARAMETERS

coupon_code

String value of coupon code for deleting coupon. Required.

DATA RETURNED

response_code

Tells the user whether or not the operation was successful.

response_message

Tells the user why the operation was or was not successful.

EXAMPLE API CALLS USING CURL

```
curl -X POST https://www.2checkout.com/api/products/delete_coupon \
-H 'Accept: application/json' -u 'username:password' \
-d 'coupon_code=discount'
```

EXAMPLE SUCCESSFUL RESPONSE

```
{
  "response_code" : "OK",
  "response_message" : "Coupon successfully deleted."
}
```

COMMON ERROR CODES

PARAMETER_MISSING

Required parameter missing: <parameter name>

PARAMETER_INVALID

Invalid value for parameter: <parameter name>

RECORD_NOT_FOUND

Unable to find record.

FORBIDDEN

Access denied to coupon.

ERRORS

If a request cannot be completed successfully, the response from the API will contain information that can be used to determine what went wrong. There are several ways to determine that the request was unsuccessful. Either the HTTP Status header will contain a status code of 400 or higher (typically 400, 404, or 500), or there will be one or more errors elements in the response. Each of these elements will contain an error code, message, and (if applicable) other informative values to assist in debugging the problem such as the offending parameter name.

Example Response:

```
{
  "code" : "PARAMETER_MISSING",
  "message" : "Required parameter missing: vendor_id",
  "parameter" : "vendor_id"
},
```

CONCLUSION

We thank you for taking the time to read this document and ask that you contact us directly with any questions. If you are developing third party software which will support 2Checkout, we invite you to become a certified shopping cart. To be certified, please contact techsupport@2co.com and provide us with access to a demo that shows proper implementation of the 2Checkout payment method using one of our system's parameter sets. Once certified, we will list your software under our recommended shopping carts section in our Knowledge Base. We will continue to update this User Guide as changes are made to our system and new features are rolled out in the future. Please let us know if you have any suggestion on this or future user guides at techsupport@2co.com

© 2Checkout.com