

WebShopApps Google Checkout Guide



Table of Contents

Google Checkout Setup and Integration.....	2
Prerequisites	2
Configuration.....	3
Installation Validations.....	3
Deficiencies.....	4
Troubleshooting Google Checkout.....	5
Debugging via Magento.....	6
Debugging via Google Checkout.....	8
Checking Callback URL.....	8

Google Checkout Setup and Integration

This section details how to setup your extension to work with Google checkout.

Prerequisites

I would firstly advise that you ensure Google Checkout is configured correctly for another shipping method such as Magento flat rate. If you cannot get flat rate to show up in Google Checkout then you will not be able to get rates from your extension. There are several things that may affect rates showing including:

1. Lack of SSL certificate, or invalid SSL configuration – Google Checkout relies on SSL being in place on the front end when it makes the callback to get the shipping rates
2. Incorrect Google Checkout configuration in Magento – please ensure Secure Callback URL is set to YES
3. Incorrect Google Checkout configuration in Google Checkout

If you need personalised support to get Google Checkout running I do offer this as a service. Contact me for pricing. I cannot do this as part of the standard extension as I hope you understand it is a time intensive process.

Configuration

To setup Google Checkout with your extension follow these steps:

1. Switch of cache (just to be on the safe side)
2. Upload your csv file into the shipping method section of your extension
3. In Google API Enable Google Checkout Shipping
4. Select all the methods you wish to make available to Google Checkout – you must make sure at least one shipping method will match by default e.g. Worldwide \$50

The screenshot shows the Magento configuration interface. On the left is a sidebar menu with categories: Tax, Checkout, Shipping Settings, Shipping Methods, Google API, PayPal Accounts, and Payment Methods. Under 'SERVICES', 'MAGENTO CORE API' is selected. Under 'ADVANCED', 'ADMIN' is selected. The main content area is titled 'Google Website Optimizer' and 'Google Checkout Shipping - Merchant Calculated'. It contains the following settings: 'Enable Merchant Calculated' is set to 'Yes'; 'Allowed Methods' lists three methods: '[premiummatrixrate] Next-Day' with a default price of 7.99, '[premiummatrixrate] 2-Day' with a default price of 5.99, and '[premiummatrixrate] Standard' with a default price of 3.99. Each method has a 'Remove' button. At the bottom of the list is an 'Add Shipping Method' button. Below this section is another header for 'Google Checkout Shipping - Carrier Calculated'.

5. Set default prices. If you don't do this and for any reason it can't get a rate, shipping will default to 0 which is undesirable.
6. Save config
7. Test functionality in front end

Installation Validations

If you wish to validate that the installation has succeeded updating the database look for the table “shipping_your-extension-name”.

If you are not getting any shipping rates check the following:

1. Do you have data in shipping_your-extension-name which matches your expectations?
2. Is the condition_name set correctly for your requirements before uploading csv (e.g. package_weight for weight filtering) if applicable?
3. Do you have correct weights setup if using weight filtering (it's not so intelligent it can tell the difference between an ounce and a pound, you control this)?
4. Have you enabled the extension and refreshed the cache?

Deficiencies

1. Does not currently support exporting of csv rates. To observe these rates please use the database directly (see table shipping_your-extension-name)
2. The csv import is taken from Magento table rate code and could be improved. In particular if the csv fails to upload sometimes you will not be notified of this. If you are not getting shipping rates out then please check the table shipping_your-extension-name to see if data was uploaded. If it was not you have an invalid csv file. The most common issue seen is that strings do not have quotes round them, which can cause issues in upload – I have seen problems when MS Excel is used to generate the csv file (as discussed by Magento on their table rates documentation). If in doubt please consult the examples.

The Google Checkout/Magento code is, in my opinion, quite a flaky piece of code, and your extension is sitting on top of this for the Google Checkout integration. So although I am confident in the extension I do not have total confidence in Google Checkout/Magento!

Troubleshooting Google Checkout

The way the Google Checkout/Magento integration works in a very simplistic form is this:

1. Magento calls Google Checkout
2. Google Checkout calls Magento
3. Magento returns shipping rates

In my opinion this handshake they have doesn't have many error checks, and any failure/recovery mechanisms built in. So if the handshake fails anywhere along the line you will see unexpected results (e.g. zero shipping rates). There is absolutely nothing my extension can do to stop this, so please bear it in mind. By setting your default shipping rates high you will mitigate it somewhat as it uses these if the handshake fails.

There are a couple of routes to debug – within Magento and within Google Checkout.

Debugging via Magento

1. Switch on Debug within the Google API->Google Checkout settings in admin.
2. Run a test on frontend where you enter Google Checkout
3. Within Phpmyadmin look at the table "googlecheckout_api_debug". You should see both a request body and a response body

The request body should look something like below:

```
<?xml version="1.0" encoding="UTF-8"?>
<merchant-calculation-callback xmlns="http://checkout.google.com/schema/2" serial-number="5747477-35235-23525-2352">
  <shopping-cart>
    <merchant-private-data>
      <quote-id><![CDATA[4]]></quote-id>
    </merchant-private-data>
    <items>
      <item>
        <item-weight value="1.0" unit="LB" />
        <tax-table-selector>none</tax-table-selector>
        <item-name>test</item-name>
        <item-description></item-description>
        <quantity>1</quantity>
        <unit-price currency="GBP">1.0</unit-price>
        <merchant-item-id>test</merchant-item-id>
        <merchant-private-item-data>
          <quote-item-id>4</quote-item-id>
        </merchant-private-item-data>
      </item>
    </items>
  </shopping-cart>
  <buyer-id>227105150286196</buyer-id>
  <calculate>
    <addresses>
      <anonymous-address id="325421987741160">
        <country-code>GB</country-code>
        <postal-code>test</postal-code>
        <city>test</city>
        <region>ESSEX</region>
      </anonymous-address>
    </addresses>
    <shipping>
      <method name="Select Shipping Method - Worldwide" />
      <method name="Select Shipping Method - Ground" />
      <method name="Select Shipping Method - Next Day" />
    </shipping>
    <tax>true</tax>
    <merchant-code-strings />
  </calculate>
  <buyer-language>en_US</buyer-language>
</merchant-calculation-callback>
```

The response body should look something like below:

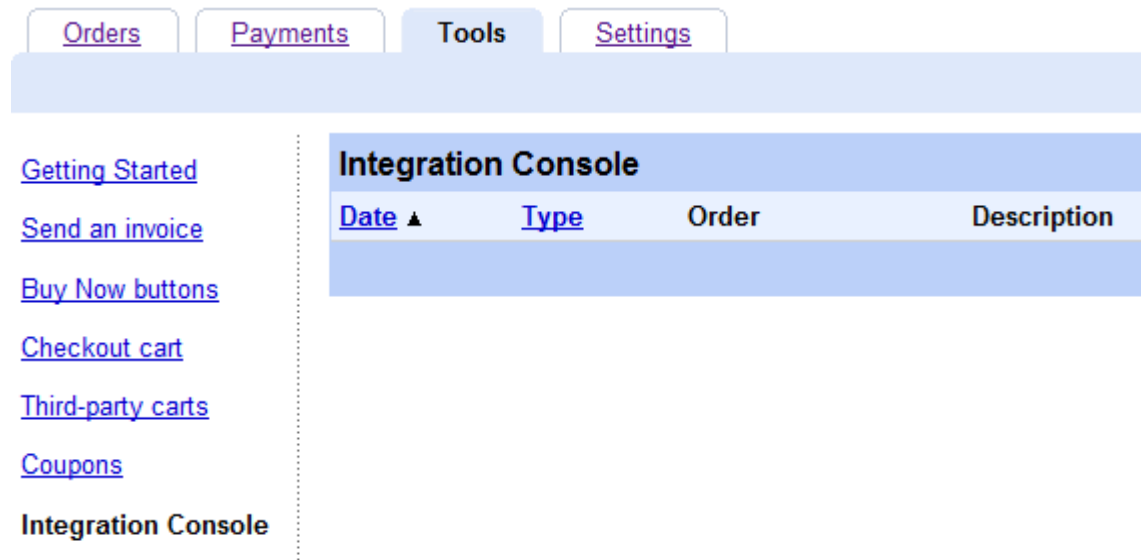
```
<?xml version="1.0" encoding="utf-8"?>
<merchant-calculation-results xmlns="http://checkout.google.com/schema/2">
  <results>
    <result shipping-name="Select Shipping Method - Worldwide" address-id="325421987741160">
      <shipping-rate currency="GBP">50</shipping-rate>
      <shippable>true</shippable>
      <total-tax currency="GBP">0</total-tax>
    </result>
    <result shipping-name="Select Shipping Method - Ground" address-id="325421987741160">
      <shipping-rate currency="GBP">0</shipping-rate>
      <shippable>false</shippable>
      <total-tax currency="GBP">0</total-tax>
    </result>
    <result shipping-name="Select Shipping Method - Next Day" address-id="325421987741160">
      <shipping-rate currency="GBP">0</shipping-rate>
      <shippable>false</shippable>
      <total-tax currency="GBP">0</total-tax>
    </result>
  </results>
</merchant-calculation-results>
```

If you aren't getting back shipping rates then check that in Magento frontend you do get back rates for the shipping address you have defined in Google Checkout. This is most commonly indicated by shippable being set to false in all rates in the response body.

If you are not seeing a response body the problem is likely to be because you haven't got Magento setup correctly, e.g. Secure Callback not switched on, SSL cert not in place.

Every time you upload a csv file I would advise you re-select the merchant shipping methods – otherwise you may see unexpected results. Strictly this is not necessary, but with this I would advise to be on safe side.

Debugging via Google Checkout



In Google Checkout itself there is an integration console. This will give details of failed callbacks to Magento, and the reasons around it. The information is not great, but should provide a pointer.

Checking Callback URL

Ensure that the callback url works. This is not really recommended unless you know what you are doing. Wget is available from the internet – do a google search.

From a command line do the following:

```
wget --post-data=foo <store_url>/googlecheckout/api/
```

You should see either a 200 OK or a 302 Found. If you see a 404 Not Found then you may have an issue with your callback.

© WebShopApps 2010. All rights reserved