

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Magento 1.x Reference

Welcome to the home page for Magento 1.x documentation for installation, configuration, developers, patches, and more. Here you'll find articles that were formerly located on <http://magentocommerce.com/knowledge-base>.

For information you don't find here—including Release Notes—see the [User Guide page](#).

Table of Contents

- [Magento 1.x system requirements](#)
- Release documentation, including installation and upgrade
 - [Magento Community Edition \(CE\) 1.8 and Enterprise Edition \(EE\) 1.13 Documentation Home](#)
 - [Magento Community Edition \(CE\) 1.9 and Enterprise Edition \(EE\) 1.14 Documentation Home](#)
- Other guides
 - [Using Redis With Magento Community Edition \(CE\) or Enterprise Edition \(EE\)](#)
 - [How to Extend the Magento REST API to Use Coupon Auto Generation](#)
 - [Magento for Developers \(8-part series\)](#)
- Magento 1.x API
 - [REST API](#)
 - [SOAP API](#)
- Other documentation
 - [How to Apply and Revert Patches](#)
 - [Discover credit card validation issue: Magento EE 1.9.1.1—1.13.1.0 and CE 1.4.2.0—1.8.1.0](#)
 - [Patches for Magento Enterprise Edition \(EE\) Versions 1.10–1.12](#)
 - [Error Using Payflow with Magento Enterprise Edition \(EE\) 1.12.0.2](#)
 - [Getting the PHP 5.4 patch for Magento Enterprise Edition \(EE\) and Community Edition \(CE\)](#)
 - [Resolving a Remote Code Execution Exploit](#)
 - [How to Install and Configure the Solr Search Engine With Magento Enterprise Edition \(EE\)](#)
 - [Information About Enterprise Edition \(EE\) Patches for Apache Solr](#)

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

System Requirements for Magento Enterprise Edition and Community Edition (Current Shipping Versions)

Magento requires a LAMP or LNMP stack.

Operating System

Linux x86-64

Web Server

- Apache 2.x
- Nginx 1.7.x

Database

- Magento EE 1.14.3 and later:
 - MySQL 5.7 (Oracle or Percona)
- Earlier Magento versions:
 - MySQL 5.6 (Oracle or Percona)

PHP

- Magento EE 1.14.4.0 and later:
 - PHP 7.2.x
- Magento CE 1.9.2 and later, Magento EE 1.14.2 - EE 1.14.3.10:
 - PHP 5.6.x
 - PHP 5.4.x
 - PHP 5.5.x
- Earlier Magento versions:
 - PHP 5.4.x
 - PHP 5.5.x

SSL

- A valid security certificate is required for HTTPS.
- Self-signed SSL certificates are not supported.

Magento can utilize the following technologies:

- [Redis](#)

Redis can be used for session or cache storage

- [Memcached](#)

memcached can be used for session or cache storage

- Apache Solr

[Solr search](#) can be used as a search provider for Magento Enterprise Edition (EE) only

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Magento Community Edition (CE) 1.8 and Enterprise Edition (EE) 1.13 Documentation Home

Table of Contents

- [Welcome](#)
- [EE 1.13 Benchmarking Guide](#)
- Release Notes
 - [Magento Community Edition \(CE\) Release Notes \(1.8 and later\)](#)
 - [Magento Enterprise Edition \(EE\) Release Notes \(1.13 and later\)](#)
- Installation
 - [Installing and Upgrading to Magento Community Edition \(CE\) and Magento Enterprise Edition \(EE\)](#)
 - [Before You Install Magento: Recommended File System Ownership and Privileges](#)
 - [After You Install Magento: Recommended File System Ownership and Privileges](#)
 - [Installing Sample Data for Magento Community Edition \(CE\)](#)
- Upgrade: [Upgrading to and Verifying Magento Community Edition \(CE\) and Enterprise Edition \(EE\)—Part 1](#)

Welcome

Welcome to the documentation home page for the Magento Enterprise Edition (EE) 1.13 and Community Edition (CE) 1.8 releases! Let's start out by telling you a little bit about them.

Magento recognizes that merchants and systems integrators require higher performance in key areas—such as product search, product browsing, and checkout. Merchants and integrators are now commonly coming to Magento with:

- Large databases (that is, more than one million products)
- Heavy traffic volume
- Frequent merchandising updates
- Extensive marketing and promotion campaigns
- Many web stores to address their customers' needs

To address these ever-growing needs, Magento systematically examined performance and scalability characteristics. Our engineers continually measured performance on a range of environments and use cases to identify and remove bottlenecks, and then conducted extensive testing to confirm improvements.

We characterized performance based on variations in deployment and business configurations to ensure improvement for a range of merchant sizes and configurations.

In this release, we focused primarily on improvements to indexing, tax calculations, caching, and checkout performance.

We're excited about the changes you'll find, including:

- Reindexing (*Enterprise Edition only*):
 - Most indexing processes now run only to update products, categories, URL redirects, and so on that have changed—eliminating the need for manual full reindexing
 - Your Magento web store is not locked at any point during reindexing
 - Reindexing is now a background process.
- Caching (*Enterprise Edition only*):
 - Full page caching now invalidates only pages that are affected by product or category changes
 - Improved cache adapter for single-host systems
 - Additional option of using [Redis NoSQL](#) for cache and session storage in multi-host deployments (recommended for new deployments)
- EE customers also get [detailed performance and tuning guidelines](#) geared toward enterprise installations
- Sales, Value Added, and Fixed Product Tax calculations (*Magento CE and EE*):
 - Major improvements in the way sales tax is calculated.
- Checkout performance (*Magento CE and EE*) achieved by:
 - Eliminating unnecessary calls to gift wrapping when loading the Shipping Method checkout step
 - Eliminating unnecessary RSS cache cleanups when RSS functionality is disabled
 - Eliminating unnecessary calls to the translation module when sending new order e-mails if the current locale is the same as the locale set in Magento
 - Improving the overall checkout process performance by loading the information for the current checkout step only
 - Improving the overall checkout process performance by loading the progress information for the current checkout step only

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Magento Community Edition (CE) 1.9 and Enterprise Edition (EE) 1.14 Documentation Home

Table of Contents

- [Welcome](#)
- [Release Notes](#)
 - [Magento Community Edition \(CE\) Release Notes \(1.9 and later\)](#)
 - [Magento Enterprise Edition \(EE\) Release Notes \(1.14 and later\)](#)
- [Developer guides](#)
 - [Responsive Email Developer's Guide for Magento EE 1.14.1 and Magento CE 1.9.1](#)
 - [Magento Community Edition \(CE\) 1.9 and Magento Enterprise Edition \(EE\) 1.14 Responsive Web Design Developer's Guide](#)
 - [Creating an iOS App for Magento MobileConnect](#)
- [Installation](#)
 - [Installing and Upgrading to Magento Community Edition \(CE\) and Magento Enterprise Edition \(EE\)](#)
 - [Before You Install Magento: Recommended File System Ownership and Privileges](#)
 - [After You Install Magento: Recommended File System Ownership and Privileges](#)
 - [Installing Sample Data for Magento Community Edition \(CE\)](#)
- [Upgrade: *Upgrading to and Verifying Magento Community Edition \(CE\) and Enterprise Edition \(EE\)—Part 1*](#)

Welcome

Welcome to the documentation home page for the Community Edition (CE) 1.9 and Magento Enterprise Edition (EE) 1.14 and releases! Let's start out by telling you a little bit about them.

Your ability to create captivating online experiences just got even better. Magento Enterprise Edition 1.14.1, now available for download from My Account, features powerful new merchandising tools and responsive design capabilities that will help you engage customers and stand out from the competition. You'll also enjoy better performance, faster technical issue resolution, and deeper insights to drive your business forward.

The new releases helps you:

- *EE only.* Merchandise product categories like never before with a new drag-and-drop tool and the ability to set up rules to automatically assign products to categories.
- Boost conversion with swatches that present richer product information to shoppers on category, search result, and product detail pages.
- Optimize sales on any device with an enhanced responsive design reference theme that now also includes responsive emails.

- Get a faster, more secure site with support for MySQL 5.6 and PHP 5.5.
- *EE only*. Diagnose technical issues faster using a new support tool that provides comprehensive information about a Magento installation.
- *EE only*. Empower larger teams by enabling many administrators to make product updates at the same time.
- Access deeper shopper insights now that Magento supports Google Universal Analytics.
- Have an even better Magento experience with more than 70 other product improvements.

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Using Redis With Magento Community Edition (CE) or Enterprise Edition (EE)

Contents

- [Overview](#)
- [Requirements for Using Redis with Magento](#)
- [Configuring Redis](#)
- [Getting Support for Redis](#)
- [For More Information](#)
- [Acknowledgment](#)

Overview

Redis is an open source, Berkeley Software Distribution (BSD) licensed, advanced key-value store that can optionally be used in Magento for backend and session storage. In fact, you can replace memcached with Redis.

Following are some of the benefits Redis provides for Magento implementations:

- Redis supports on-disk save and master/slave replication. This is a powerful feature not supported by memcached. Replication enables high availability by eliminating a single point of failure.
- Redis can be used for PHP session storage.
- Redis provides much better eviction control and its backend is written with eviction support in mind.
- Redis supports multiple databases that use the same server instance so you can use different databases for the Magento cache, full page cache (*EE only*), and sessions without starting many processes listening on different ports.
- Redis supports compression libraries `gzip`, `lzf`, and `snappy`. lzf and snappy are much faster than gzip.
- Limits the number of concurrent lock requests before an [HTTP 503 \(Service Unavailable\)](#) error is returned.

Requirements for Using Redis with Magento

The following Magento editions support Redis session and backend caching:

- Enterprise Edition (EE) 1.13 and later
- Community Edition (CE) 1.8

The preceding Magento editions support Redis server version 2.6.9 and later available from [redis.io](#).

In addition, you can optionally use the [Redis extension for PHP](#) version 2.2.3 or later if you're using Redis for backend caching; however, Magento works without this extension.

Configuring Redis

To use Redis with Magento, you must configure Magento to use Redis and you must install and configure the Redis server. These tasks are discussed in the following sections:

- [Configuring Magento To Use Redis](#)
- [Installing and Configuring the Redis Server](#)

Configuring Magento To Use Redis

To use Redis with Magento, you only need to install and configure the Redis server. The integration between Redis and Magento is already included with Magento CE 1.8 and EE 1.13 and later versions. All you need to do is configure it.

Important: The Cm_RedisSession module in CE 1.8 is disabled by default. Magento disables the module to avoid unnecessary connection tries to Redis when you choose to use file, database, or a different session storage method.

To enable Magento to use Redis, perform the following tasks:

1. Enable the Cm_RedisSession module.
 - a. Open *magento-install-dir/app/etc/modules/Cm_RedisSession.xml* in a text editor.
 - b. Change the value of <active> to true.
 - c. Save your changes to *Cm_RedisSession.xml* and exit the text editor.
2. Modify *magento-install-dir/app/etc/local.xml*.
For configuration information, see the sample provided with Magento in *magento-install-dir/app/etc/local.xml*. additional and also see the [Readme \(session\)](#) and [Readme \(backend\)](#).
3. Flush the Magento cache in any of the following ways:
 - If you have access to the file system as the owner of the files in the Magento installation directory, change to that directory and enter `rm -rf var/cache`.
 - Log in to the Admin Panel as an administrator. Click **System > Cache Management**, then click **Flush Magento Cache** at the top of the page.

Installing and Configuring the Redis Server

Get [Redis server](#) version 2.6.9 or later and configure it according to [their documentation](#).

You can optionally install the [Redis extension for PHP](#) version 2.2.3 or later as well, but Magento functions without it.

Getting Support for Redis

Important: Colin Mollenhour, the original author of Redis, does *not* provide support for Magento implementations. You can get support for Magento implementations in the following ways:

| MAGENTO EDITION | HOW TO GET HELP |
|---------------------------|---|
| Magento CE and Magento EE | <ul style="list-style-type: none">• Magento support forums• Twitter (@ask_magento) |
| Magento CE | Submit bug reports |
| Magento EE only | Submit bug reports |

For More Information

For more information about using Redis with Magento, see:

- [Magento Expert Consulting Group \(ECG\) article](#).
- [Redis server documentation](#).

Acknowledgment

Magento acknowledges the contributions of Colin Mollenhour in providing the code for the Magento implementation of Redis.

22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020.

Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

How to Extend the Magento REST API to Use Coupon Auto Generation

Table of Contents

- [Overview](#)
- [System Requirements](#)
- [Generating Coupon Codes](#)
- [Extending Magento's REST API to Include Coupon Auto-Generation](#)
- [Securing the Coupon AutoGen API](#)
- [Testing the Coupon AutoGen API](#)
- [Troubleshooting Suggestions](#)
- [Next Steps](#)

Overview

Customers of traditional stores and online web stores love coupons. Typically, a merchant sends coupons to customers who input them when checking out. The coupon saves the customer money and hopefully entices the customer to visit the store more frequently. In addition, the merchant can track coupon codes to individual customers to target market those customers.

Coupon Auto Generation and the Magento REST API

Magento CE 1.7 introduced a new method of creating coupon codes—*auto generation*. Auto generating coupons means Magento programmatically creates several coupon codes at one time quickly and easily. However, if Magento generates the coupon codes, you'd have to manually

distribute them to customers.

Magento's REST API is extensible and can easily be called by an outside program to auto generate coupon codes. You can use this feature, for example, to e-mail coupon codes to your top 100 customers.

No programming is necessary to implement the extension module discussed in this guide; however, basic familiarity with [Magento modules](#) and PHP programming is desirable.

The Coupon AutoGen API enables any authorized external program to instruct Magento to:

- Auto-generate the specified number of coupon codes
- Return these codes to the calling program—simulated in this guide using a simple .php file

Implementation Details

This guide discusses how to use coupon auto generation and a web service to dynamically call the Magento REST API to generate a series of codes. The web service instantiates the underlying Magento sales rule (`salesrule/rule`) coupon code generator and creates a pool of new codes. These codes returned to the caller as a JSON string.

For more information about the Magento REST API, see [Introduction to REST API](#). To extend the REST API to add a web service for generating and retrieving coupon codes, this guide discusses the following:

- Setting up an auto generated coupon rule
 - Creating a module to use a REST based API extension
- This guide gives you all the files to create the module; no programming is necessary. There are four module configuration .xml files and one .php file to create the web service.
- Creating a user for the OAuth access to the new service
 - Testing the web service from a separate page

System Requirements

To implement and test the Coupon AutoGen API, you must have all of the following:

- Magento Community Edition (CE) 1.7 or later on Ubuntu.
- Magento Enterprise Edition (EE) 1.12 or later on Ubuntu.
- [pecl OAuth 1.0a extension](#) which you install as discussed in [Installing OAuth](#). Magento CE 1.7 and later support the [OAuth 1.0a specification](#).
- [*Optional.* phpmyadmin](#), which makes it easier to view and manipulate the Magento database. You can use `phpmyadmin` for convenience to get the OAuth key and shared secret later in this guide.



Note: Although Magento supports other Linux operating systems, Ubuntu is the only one discussed in this guide. Consult an appropriate reference for equivalent commands on other Linux operating systems.

Installing OAuth

This section discusses how to install the [pecl OAuth extension](#) on Ubuntu. Consult the [pecl.php.net documentation](#) for installation instructions on other operating systems.

The pecl OAuth extension requires the PEAR installer. The following sections discuss the installation in detail:

- [*Creating a phpinfo File*](#)

- [Installing the OAuth Packages](#)
- [Confirming that OAuth Installed Successfully](#)



Note: You must perform the tasks discussed in this section as a user with **root** privileges.

Creating a phpinfo File

Before you begin, create a `phpinfo.php` file, if you have not already done so, to determine if you already have OAuth running.

To create `phpinfo.php`:

1. Open a command prompt window and connect to your Magento server.
2. Create a file named `phpinfo.php` anywhere on the web server's docroot:

```
<?php
// Show all information, defaults to INFO_ALL
phpinfo();
```

3. Start a web browser and enter the following URL in its address or location field:

```
http://host-or-ip[:port]/path-to-phpinfo/phpinfo.php
```

For example, if your Magento instance hostname is `www.example.com` and you put `phpinfo.php` in the web server's docroot, enter:

```
http://www.example.com/phpinfo.php
```

4. Search the resulting output for **OAuth**.

The following figure shows an example of OAuth being properly set up.

OAuth

| OAuth support | enabled |
|-------------------------------|--|
| PLAINTEXT support | enabled |
| RSA-SHA1 support | enabled |
| HMAC-SHA1 support | enabled |
| Request engine support | php_streams |
| source version | \$Id: oauth.c 325799 2012-05-24 21:07:51Z jawed \$ |
| version | 1.2.3 |

If the preceding does *not* display, OAuth is not set up so continue with the next section.

If OAuth is already installed, continue with [Defining a Magento Coupon Code Generation Rule](#).

Installing the OAuth Packages

The pecl OAuth extension requires both PEAR (which enables you to install the package) and `libpcre3-dev`, which enables the OAuth package to be compiled.

To install the packages and confirm that OAuth is enabled:

1. Enter the following commands in the order shown:

```
apt-get install php-pear
apt-get install libpcre3-dev
pecl install oauth
```



Note: Remember that you must perform the tasks discussed in this section as a user with **root** privileges.

Wait while each package is installed. The message **Build process completed successfully** displays to indicate OAuth installed successfully.



Note: If Error: `make` failed displays after you enter `pecl install oauth`, see [OAuth Package Installation Error: `make` failed](#).

If the following displays, you must edit your `php.ini` file to find the OAuth library:

```
configuration option "php_ini" is not set to php.ini location  
You should add "extension=oauth.so" to php.ini
```

2. Open `php.ini` in a text editor.

If you're not sure where it's located look in the `phpinfo.php` page output. Add the following anywhere in `php.ini`:

```
[OAuth]  
extension=oauth.so
```

3. Save your changes to `php.ini` and exit the text editor.

4. Enter the following command to restart the Apache web server.

```
service apache2 restart
```

5. Continue with the next section.

Confirming that OAuth Installed Successfully

If your `phpinfo.php` page is still open in a web browser, press Control+R to force a refresh; otherwise, enter the URL shown in [Creating a phinfo File](#) to view it.

The following figure shows an example of a properly set up OAuth extension.

OAuth

| OAuth support | enabled |
|------------------------|--|
| PLAINTEXT support | enabled |
| RSA-SHA1 support | enabled |
| HMAC-SHA1 support | enabled |
| Request engine support | php_streams |
| source version | \$Id: oauth.c 325799 2012-05-24 21:07:51Z jawed \$ |
| version | 1.2.3 |



Note: Do not continue until you know that OAuth installed successfully.

Defining a Magento Coupon Code Generation Rule

To define a Magento coupon code generation rule:

1. Log in to the Magento Admin Panel as an administrator.

2. Click **Promotions > Shopping Cart Price Rules**.
 3. On the Shopping Cart Price Rules page, click **Add New Rule** (in the upper-right corner of the page).
- The General Information page displays as follows.

The screenshot shows the 'General information' section of a configuration interface. It includes fields for Rule Name (set to 'Generate Coupons'), Description (empty), Status (set to 'Active'), Websites (set to 'Main Website'), Customer Groups (set to 'NOT LOGGED IN'), Coupon (set to 'Specific Coupon'), and various numerical and date inputs for uses per coupon, uses per customer, and date ranges. A checkbox for 'Use Auto Generation' is checked, with a note explaining it generates multiple coupon codes if selected and saved. Other fields include Priority (0) and Public in RSS Feed (No).

4. Enter the following information.

| ITEM | DESCRIPTION |
|-----------------|---|
| Rule Name | Enter Generate Coupons. |
| Description | Enter an optional description of the rule, such as Rule that generates a sequence of coupon codes. |
| Status | From the list, click Active . |
| Websites | Click the websites on which you want the coupons to display. Hold down the Shift key and click the names of all items to select them. |
| Customer Groups | Hold down the Shift key and click the names of all items to select them. |
| Coupon | Click Specific Coupon |

| | |
|--------------------------|---|
| Coupon Code | Leave the field blank. |
| CE only. Auto Generation | Select the Use Auto Generation checkbox. |
| CE only. Uses per Coupon | Enter 10. |
| Uses per Customer | Enter 1. |
| From Date | Select today's date. |
| To Date | Select any date in the future. |
| Priority | Enter 0. |
| Public In RSS Feed | Click No . |

5. In the upper-right corner of the page, click **Save and Continue Edit**.

The message **The rule has been saved** displays at the top of the page to indicate that Magento successfully saved the rule you just created.

6. In the left navigation bar, click **Actions**.

The Actions page displays as follows.

The screenshot shows the 'Edit Rule 'Generate Coupons'' page. On the left, a sidebar menu lists: Rule Information, Conditions, **Actions**, Labels, and Manage Coupon Codes. The main content area is titled 'Edit Rule 'Generate Coupons''. It contains two main sections: 'Update prices using the following information' and 'Apply the rule only to cart items matching the following conditions (leave blank for all items)'. Under 'Update prices', 'Apply' is set to 'Percent of product price discount' with a value of '10'. 'Discount Qty Step (Buy X)' is set to '1'. Under 'Apply the rule only to cart items', it says 'If ALL of these conditions are TRUE:' followed by a condition entry field containing a green circle icon.

7. Enter the following information.

| ITEM | DESCRIPTION |
|------------------------------------|---|
| Apply | From the list, click Percent of product price discount . |
| Discount Amount | Enter 10. |
| Maximum Qty Discount is Applied To | Enter 1. |
| Discount Qty Step (Buy X) | Enter 1. |
| Apply to Shipping Amount | From the list, click No . |
| Free Shipping | From the list, click No . |
| Stop Further Rules Processing | From the list, click Yes . |

- Click **Save** (in the upper-right corner of the page).

The message **The rule has been saved** displays to indicate that Magento saved the rule action options you just entered. Notice that this page now has a row for the Generate Coupons rule you just defined.

A sample follows.



The screenshot shows a grid titled "Shopping Cart Price Rules". At the top right are buttons for "Add New Rule", "Reset Filter", and "Search". Below the title is a header row with columns: ID, Rule Name, Coupon Code, Date Start, Date Expire, Status, Website, and Priority. The first column contains a small icon. The second column shows the rule name "Generate Coupons". The third column is empty. The fourth column has date pickers set to "From: Oct 24, 2012" and "To: Nov 30, 2012". The fifth column shows "Status: Active". The sixth column shows "Website: Main Website". The seventh column shows "Priority: 0". At the bottom left of the grid, the number "3" is circled in red.

- Write down the rule ID (circled in red in the preceding figure). You will use this value later in this guide.

Generating Coupon Codes

Now that you've created a rule, this section discusses how to use the rule to manually generate a sequence of coupon codes.

- In the Shopping Cart Price Rules page, click the name of the rule you just created (**Generate Coupons**).
- In the left navigation bar, click **Manage Coupon Codes**.
- On the Coupons Information page, enter the following information:

| ITEM | DESCRIPTION |
|-------------------------|-----------------------------|
| Coupon Qty | Enter 3. |
| Code Length | Enter 12. |
| Code Format | Click Alphanumeric . |
| Coupon Prefix | Enter TEST-. |
| Coupon Suffix | Enter -TEST. |
| Dash Every X Characters | Enter 0. |

- Click **Generate**.

Magento displays the three coupon codes you created in the Coupon Code section as the following figure shows.

5. Click **Save**.

Extending Magento's REST API to Include Coupon Auto-Generation

In the preceding section, you created a Shopping Cart Price Rule named Generate Coupons that manually generates a set of coupon codes. To use those codes, you could export them to a file, and then import them into any external program you want; however, this is a time-consuming procedure!

Fortunately, you can automate this process by adding a *coupon code auto-generate* API to Magento's existing REST API. Using this API, an external program can automatically get the coupon codes it needs.

The following sections discuss how to extend Magento's REST API to include the Coupon AutoGen API:

- [Disabling the Magento Cache](#)
- [Creating Configuration Files](#)
- [Setting Permissions on the Configuration Files](#)

Disabling the Magento Cache

While you're implementing the Coupon AutoGen API, you must disable Magento's caching so Magento will find and use your new code immediately.

To disable the cache:

1. In the Magento Admin Panel, click **System > Cache Management**.

The Cache Storage Management page displays as follows.

2. Click **Select All** (on the upper-left of the page).

A check mark displays next to each Magento cache in the list.

3. From the **Actions** list, click **Disable**.

4. Click **Submit**.

Magento disables the selected caches, replacing the green ENABLED status indicators with red DISABLED indicators.

5. Click **Flush Magento Cache** and wait for the cache to be flushed.

6. Log out of the Magento Admin Panel.

Creating Configuration Files

This section discusses how to create a module (also referred to as an *extension*). The module consists of configuration files that create a web service that extends the Magento REST API to take input from an external program. This program uses **HTTP POST** and OAuth calls to auto-generate coupon codes.

In this guide, the external program is a PHP script; however, it could be any application that uses OAuth and REST calls.

For more information about Magento module development, see [developer documentation on Magento Connect](#).



Note: It is up to you to determine ownership and permissions. This guide assumes you create files and directories as **root** and change the permissions appropriately later. Consult an IT administrator if you're not sure how to proceed.

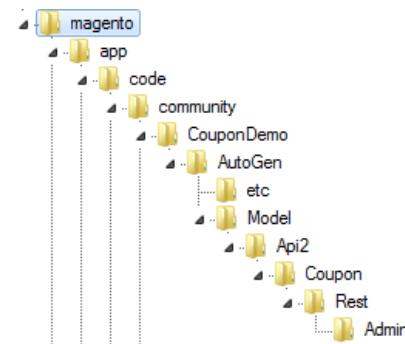
1. Log in to your Magento server and create the following directories:

```
magento-install-dir/app/code/community/CouponDemo  
magento-install-dir/app/code/community/CouponDemo/AutoGen  
magento-install-dir/app/code/community/CouponDemo/AutoGen/etc  
magento-install-dir/app/code/community/CouponDemo/AutoGen/Model  
magento-install-dir/app/code/community/CouponDemo/AutoGen/Model/Api2  
magento-install-dir/app/code/community/CouponDemo/Model/Api2/Coupon  
magento-install-dir/app/code/community/CouponDemo/Model/Api2/Coupon/Rest  
magento-install-dir/app/code/community/CouponDemo/Model/Api2/Coupon/Rest/Admin
```

For example, if Magento is installed in `/var/www/magento`, create the following directories:

```
/var/www/magento/app/code/community/CouponDemo  
/var/www/magento/app/code/community/CouponDemo/AutoGen/etc  
/var/www/magento/app/code/community/CouponDemo/Model/Api2/Coupon/Rest/Admin
```

The following figure shows an example.



Important:

- You must create the files and directories *exactly* as shown; directory names are

case-sensitive.

- Do not change any values in the configuration files discussed in this section.
- When you create your configuration files, leave *no white space* at the beginning of the files. Leading white space might cause errors when Magento reads the files and might prevent the coupon demonstration module from working.

2. Change to the *magento-install-dir/app/etc/modules* directory.
3. In that directory, use a text editor to create your module declaration file. This file must be named *CouponDemo_AutoGen.xml* and have the following contents.

```
<?xml version="1.0"?>
<config>
    <modules>
        <CouponDemo_AutoGen>
            <active>true</active>
            <codePool>community</codePool>
        </CouponDemo_AutoGen>
    </modules>
</config>
```

4. Save your changes to *CouponDemo_AutoGen.xml* and exit the text editor.
5. Change to the *magento-install-dir/app/code/community/CouponDemo/AutoGen/etc* directory.
6. Create a file named *config.xml* with the following contents.

```
<?xml version="1.0"?>
<config>
    <modules>
        <CouponDemo_AutoGen>
            <version>0.1.0</version>
        </CouponDemo_AutoGen>
    </modules>
    <global>
        <models>
            <autogen>
                <class>CouponDemo_AutoGen_Model</class>
            </autogen>
        </models>
    </global>
</config>
```

7. Create a file named *api2.xml* with the following contents.

```
<config>
    <api2>
        <resource_groups>
            <autogen translate="title" module="CouponDemo_AutoGen">
                <title>CouponDemo API</title>
                <sort_order>10</sort_order>
            </autogen>
        </resource_groups>
        <resources>
            <autogen translate="title" module="CouponDemo_AutoGen">
```

```

<group>autogen</group>
<model>autogen/api2_coupon</model>
<title>Coupon Code Auto Generation</title>
<sort_order>10</sort_order>
<privileges>
    <admin>
        <create>1</create>
        <retrieve>1</retrieve>
    </admin>
</privileges>
<attributes>
    <coupon_id>Coupon ID</coupon_id>
    <code>Code</code>
    <qty>Quantity</qty>
    <length>Length</length>
    <format>Format</format>
</attributes>
<routes>
    <route>
        <route>/coupondemo/rules/:rule_id/codes</route>
        <action_type>collection</action_type>
    </route>
</routes>
<versions>1</versions>
</autogen>
</resources>
</api2>
</config>

```

8. Save your changes to `api2.xml` and exit the text editor.
9. Change to the `Magento-install-dir/app/code/community/CouponDemo/AutoGen/Model/Api2` directory.
10. Create a file named `Coupon.php` with the following contents.

```

<?php
class CouponDemo_AutoGen_Model_Api2_Coupon extends Mage_Api2_Model_Resource
{
}

```

11. Save your changes to `Coupon.php` and exit the text editor.
12. Change to the `Magento-install-dir/app/code/community/CouponDemo/AutoGen/Model/Api2/Coupon/Rest/Admin` directory.
13. Create a file named `V1.php` with the following contents.

```

<?php
/* Coupon AutoGen REST API
 *
 * @category CouponDemo
 * @package CouponDemo_AutoGen
 * @author Chuck Hudson (used with permission). For more recipes, see Chuck's book http://shop.oreilly.com/product/0636920023968.do
 */
class CouponDemo_AutoGen_Model_Api2_Coupon_Rest_Admin_V1 extends CouponDemo_AutoGen_Model_Api2_Coupon
{

```

```

    /**
     * Generate one or more coupon codes using the Generate Coupons rule defined in Magento.
     * Expected parameters are:
     * {
     *     'qty': int, - number of coupon codes to instruct Magento to generate
     *     'length': int, - length of each generated coupon code
     *     'format': string, - alphanumeric (for alphanumeric codes), alpha (for alphabetical codes), and num (for numeric codes)
     * }
     *
     * @param array $couponData
     * @return string|void
     */
    protected function _create($couponData)
    {
        $ruleId = $this->getRequest()->getParam('rule_id');
        $couponData['rule_id'] = $ruleId;
        $rule = $this->_loadSalesRule($ruleId);
        // Reference the MassGenerator on this rule.
        /** @var Mage_SalesRule_Model_Coupon_Massgenerator $generator */
        $generator = $rule->getCouponMassGenerator();
        // Validate the generator
        if (!$generator->validateData($couponData)) {
            $this->_critical(Mage::helper('salesrule')->__(('Coupon AutoGen API: Invalid parameters passed in.')),
                Mage_Api2_Model_Server::HTTP_BAD_REQUEST);
        } else {
            // Set the data for the generator
            $generator->setData($couponData);
            // Generate a pool of coupon codes for the Generate Coupons rule
            $generator->generatePool();
        }
    }
    /**
     * Retrieve list of coupon codes.
     *
     * @return array
     */
    protected function _retrieveCollection()
    {
        $ruleId = $this->getRequest()->getParam('rule_id');
        $rule = $this->_loadSalesRule($ruleId);
        /** @var Mage_SalesRule_Model_Resource_Coupon_Collection $collection */
        $collection = Mage::getResourceModel('salesrule/coupon_collection')
        ;
        $collection->addRuleToFilter($rule);
        $this->_applyCollectionModifiers($collection);
        $data = $collection->load()->toArray();
        return $data['items'];
    }
    /**
     * Load sales rule by ID.
     *
     * @param int $ruleId
     * @return Mage_SalesRule_Model_Rule
     */
    protected function _loadSalesRule($ruleId)
    {
        if (!$ruleId) {

```

```

        $this->_critical(Mage::helper('salesrule')
            ->__('Rule ID not specified.'), Mage_Api2_Model_Server::HTTP_BAD_REQUEST);
    }
    $rule = Mage::getModel('salesrule/rule')->load($ruleId);
    if (!$rule->getId()) {
        $this->_critical(Mage::helper('salesrule')
            ->__('Rule was not found.'), Mage_Api2_Model_Server::HTTP_NOT_FOUND);
    }
    return $rule;
}
}

```

14. Save your changes to V1.php and exit the text editor.

Setting Permissions on the Configuration Files

File permissions and ownership are important for any Linux application. Magento provides general guidelines for permission and ownership although following them are not a requirement for this guide. The configuration files and directories can be owned by `root` or other users and it won't prevent the procedures discussed in this guide from completing successfully.

Consult your network administrator if you are not sure how to set file permissions and ownership. The procedure that follows is a suggestion only.

The Magento guidelines discussed in the following procedure are taken from this Magento Wiki article and set the following:

- File and directory ownership set to `your-login-name:apache-user-group`. If you're not sure which group owns Apache processes, enter the command `ps -ef | grep apache2`. The following procedure assumes it is `www-data`.
- File permissions set to 644.
- Directory permissions set to 755.

To optionally set permissions and ownership according to Magento guidelines:

1. As a user with `root` privileges, enter the following commands in the order shown to change ownership of the files and directories you created as discussed in this guide:

```

cd magento-install-dir/app/code/community
chown -R your-login-name:www-data CouponDemo
find . -type f -exec chmod 644 {} +
find . -type d -exec chmod 755 {} +

```

2. As a user with `root` privileges, enter the following commands in the order shown to change the permissions and ownership of `CouponDemo_AutoGen.xml`

```

cd magento-install-dir/app/etc/modules
chown your-login-name:www-data CouponDemo_AutoGen.xml
chmod 644 CouponDemo_AutoGen.xml

```

Securing the Coupon AutoGen API

For security reasons, Magento allows only authorized external programs to call the Magento REST API.

The following sections discuss how to enable the test script (discussed in the next section) to call the Coupon AutoGen API:

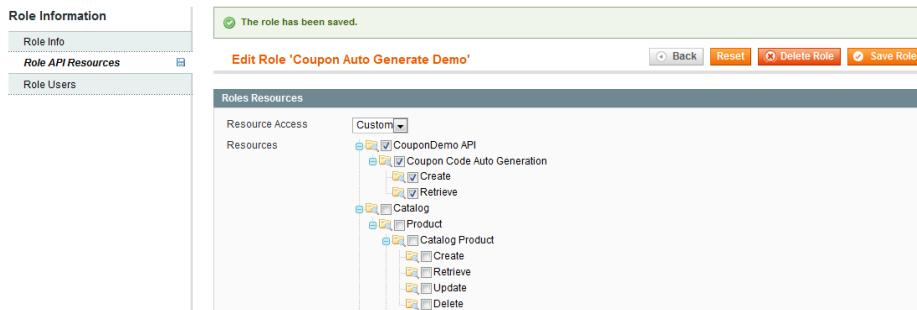
- [Creating a REST role for the Coupon AutoGen API](#)
- [Adding Users to the Coupon Auto Generate Demo Role](#)
- [Setting up REST Attributes for the Coupon AutoGen API](#)
- [Creating an OAuth Consumer for Your Test Script](#)

Creating a REST role for the Coupon AutoGen API

To use the Magento Admin Panel to create a role for the Coupon AutoGen API:

1. Log in to the Magento Admin Panel as an administrator.
2. Click **System > Web Services > REST - Roles**.
3. On the REST—Roles page, click **Add Admin Role**.
4. In the **Role Name** field, enter **Coupon Auto Generate Demo**.
5. Click **Save Role**.
6. In the left navigation bar, click **Role API Resources**.
The Role Resources page contains a hierarchical list of resources to which you can grant or deny the Coupon Auto Generate Demo role access.
7. From the **Resource Access** list, click **Custom**.
8. Select the checkbox next to the node labeled **CouponDemo API**.

Magento automatically checks the child checkboxes as the following figure shows.



9. Click **Save Role**.

Magento saves the resource API permissions you granted to the Coupon Auto Generate Demo REST role.

The Coupon Auto Generate Demo role now has permission to use the Coupon AutoGen API.

Adding Users to the Coupon Auto Generate Demo Role

Now that you have a role, you must add users to give them permission to call the Coupon AutoGen API as follows:

1. In the left navigation bar, click **Role Users**.

2. Click **Reset Filter** (in the upper-right corner of the page).

The page displays all registered users as the following figure shows.

| ID | User Name | First Name | Last Name |
|----|----------------------|------------|-----------|
| 2 | xComMagentoCEExample | Magento | User |
| 1 | admin | Store | Owner |

3. Select the checkbox next to each user to grant the user privileges to access the resources available to the Coupon Auto Generate Demo REST role—that is, permission to call the Coupon AutoGen API.



Note: If a warning dialog box displays, click **OK** to dismiss it. This warning is not relevant when adding users to REST roles.

4. When you're done, click **Save Role**.

The specified user(s) can now grant an external program the right to call the Coupon AutoGen API.

Setting up REST Attributes for the Coupon AutoGen API

This section discusses how to enable any user with a REST Admin role to use the Coupon AutoGen API.

To set REST attributes for the REST Admin role:

1. Click **System > Web Services > REST - Attributes**.
2. On the REST Attributes page, under User Type, click **Admin**.
3. In the User Type Resources section, from the **Resource Access** list, click **Custom**.
4. Select the **CouponDemo API** checkbox.

Doing so selects all the child checkboxes, as the following figure shows.

5. Click **Save**.

Any user with the REST Admin role can now read from and write to the Coupon AutoGen API.

Creating an OAuth Consumer for Your Test Script

This section discusses how to create a consumer so you can test the Coupon AutoGen API before you deploy it in a production system. After successfully testing the API, you can remove this user.

1. In the Magento Admin Panel, click **System > Web Services > REST - OAuth Consumers**.
2. Click **Add New** (in the upper-right corner of the page).

The **New Consumer** page displays as the following figure shows.

New Consumer

Back Reset Save Save and Continue Edit

| Consumer Information | |
|-----------------------|----------------------------------|
| Name * | Coupon AutoGen Test Driver |
| Key * | m156fly2bp5wbp4b3ilizvz83fl6j |
| Secret * | r06hhbguckro4ikr4xua7krv1z1fgrd5 |
| Callback URL | |
| Rejected Callback URL | |

3. In the **Name** field, enter Coupon AutoGen Test Driver.
4. Leave the other fields blank.
5. *Write down* the values displayed in the **Key** and **Secret** text boxes.



Note: The key and secret values are stored in the Magento database in the table `oauth_consumer`. It might be more convenient for you to use `phpmyadmin` or database tools to retrieve them from the database after you save the role.

You must include these values in the test script you will write in the next section. The script uses these values to identify itself to Magento.

6. Click **Save** (in the upper-right corner of the page).
7. Log out of the Magento Admin Panel.

Testing the Coupon AutoGen API

This section discusses how to create a simple PHP file that acts as an external program and, with permissions you granted the OAuth consumer, enables the program to use the [HTTP POST](#) method to auto generate coupon codes.

You can use *any* type of OAuth/REST call, in fact, such as using the Firefox REST Client plug-in as discussed [here](#).

The following sections discuss how to create and run the test script:

- [Creating the Test Script](#)
- [Running the Test Script](#)

Creating the Test Script

The test script you create calls the Coupon AutoGen API, thereby causing Magento to generate the specified coupon codes and return them to the caller (`rest_test.php`) in the form of a JSON-encoded string. Finally, the server responds to the browser's request with an HTML page containing the generated coupon codes.

The PHP code that follows:

- Uses your OAuth `consumerKey` and `consumerSecret` to set up the OAuth client.
- Defines data to pass to the array `$couponGenerationData`. The data should include the following:
 - The rule ID (listed next to the rule in the rule list in the Magento Admin Panel)
 - The number of codes desired (in this case, two)
 - The length of the codes
 - The format to be used (in this case, alphanumeric)

- The array of parameters is then passed in the `fetch` command with the `resourceUrl` for the web service call.

Finally, the server responds to the browser's request with an HTML page containing the generated coupon codes.

To create the test script, named `rest_test.php`:

- Create the file `magento-install-dir/rest_test.php` with the following contents.

```
<?php
/*****
File name: rest_test.php
Description:
A PHP test script that calls the Coupon AutoGen extension
to Magento's REST API.
The Coupon AutoGen API takes:
-- the rule ID of the "Generate Coupons" rule to execute
-- the number of coupon codes to generate
-- the length of each coupon code
-- the format of each coupon code
The API returns the generated coupon codes, in JSON-encoded form
*****/
// Replace <> below with the key and secret values generated for the C
oupon AutoGen Test Driver
$consumerKey = '<>YOUR CONSUMER KEY>'; // from Admin Panel's "REST - OAuth
Consumers page"
$consumerSecret = '<>YOUR CONSUMER SECRET>'; // from Admin Panel's "REST -
OAuth Consumers page"

// Set the OAuth callback URL to this script since it contains the logic
// to execute *after* the user authorizes this script to use the Coupon Aut
oGen API
$callbackUrl = "http://<>host-or-ip:port>/<>path>/rest_test.php";

// Set the URLs below to match your Magento installation
$temporaryCredentialsRequestUrl = "http://<>host-or-ip:port>/<>path>/oauth
/initiate?oauth_callback=" . urlencode($callbackUrl);
$adminAuthorizationUrl = 'http://<>host-or-ip:port>/<>path>/admin/oauth_a
uthorize';
$accessTokenRequestUrl = 'http://<>host-or-ip:port>/<>path>/oauth/token';
$apiUrl = 'http://<>host-or-ip:port>/<>path>/api/rest';

session_start();
if (!isset($_GET['oauth_token']) && !isset($_SESSION['state']) && $_SESSION[
'state'] == 1) {
    $_SESSION['state'] = 0;
}
try {
    $authType = ($_SESSION['state'] == 2) ? OAUTH_AUTH_TYPEORIZATION :
OAUTH_AUTH_TYPE_URI;
    $oauthClient = new OAuth($consumerKey, $consumerSecret, OAUTH_SIG_METHO
D_HMACSHA1, $authType);
    $oauthClient->enableDebug();
    if (!isset($_GET['oauth_token']) && !$_SESSION['state']) {
        $requestToken = $oauthClient->getRequestToken($temporaryCredentials
RequestUrl);
        $_SESSION['secret'] = $requestToken['oauth_token_secret'];
        $_SESSION['state'] = 1;
        header('Location: ' . $adminAuthorizationUrl . '?oauth_token=' . $r
equestToken['oauth_token']);
        exit;
    } else if ($_SESSION['state'] == 1) {
```

```

        $oauthClient->setToken($_GET['oauth_token'], $_SESSION['secret']);
        $accessToken = $oauthClient->getAccessToken($accessTokenRequestUrl)
;

        $_SESSION['state'] = 2;
        $_SESSION['token'] = $accessToken['oauth_token'];
        $_SESSION['secret'] = $accessToken['oauth_token_secret'];
        header('Location: ' . $callbackUrl);
        exit;
    } else {
        // We have the OAuth client and token. Now, let's make the API call
        .
        $oauthClient->setToken($_SESSION['token'], $_SESSION['secret']);
        // Set the array of params to send with the request
        $ruleId = <<RULE_ID>>; // Set to the rule ID of the Generate Coupon
        s rule
        $couponGenerationData = array();
        $couponGenerationData['qty'] = 2; // Number of coupons codes to cre
        ate
        $couponGenerationData['length'] = 7; // Length of each coupon code
        // Options for format include:
        // alphanum (for alphanumeric codes), alpha (for alphabetical codes
        ), and num (for numeric codes)
        $couponGenerationData['format'] = "alphanum"; // Use alphanumeric f
        or the coupon code format
        // Generate coupon codes via POST
        $resourceUrl = "$apiUrl/coupondemo/rules/{$ruleId}/codes";
        $oauthClient->fetch($resourceUrl, json_encode($couponGenerationData
), OAUTH_HTTP_METHOD_POST, array(
            'Accept' => 'application/json',
            'Content-Type' => 'application/json',
));
        // Retrieve list of created coupons via GET
        $collectionFilters = array('limit' => $couponGenerationData['qty'],
        'order' => 'coupon_id', 'dir' => 'dsc');
        $oauthClient->fetch($resourceUrl, $collectionFilters, OAUTH_HTTP_ME
THOD_GET, array(
            'Accept' => 'application/json',
            'Content-Type' => 'application/json',
));
        $coupons = json_decode($oauthClient->getLastResponse(), true);
        // Display the newly generated codes to demonstrate that the Coupon
        AutoGen API works
        // In reality, you might put these codes in emails to customers, st
        ore them in a database, etc.
        echo "New coupon codes:
";
        foreach ($coupons as $coupon) {
            echo " --> " . $coupon['code'] . "
";
        }
    }
} catch (OAuthException $e) {
    print_r($e->getMessage());
    echo "
";
    print_r($e->lastResponse);
}

```

2. The following table discusses the values you must change.

| STRING TO CHANGE | HOW TO CHANGE IT |
|------------------------------|--|
| <>YOUR CONSUMER KEY>> | Coupon AutoGen Test Driver OAuth consumer's key. You can view this in the Admin Panel: System > Web Services > REST - OAuth Consumers or you can get the value from the oauth_consumer table in the Magento database. |
| <>YOUR CONSUMER SECRET>> | Coupon AutoGen Test Driver OAuth consumer's secret. You can view this in the Admin Panel: System > Web Services > REST - OAuth Consumers or you can get the value from the oauth_consumer table in the Magento database. |
| <>host-or-ip:port>>/<>path>> | Your Magento instance's fully qualified hostname or IP address and port, if you are using a port other than 80, and the path to your Magento installation. If you are running Magento on localhost, enter 127.0.0.1 For example, if your Magento server's hostname is www.example.com, running on port 80, and Magento is installed at /var/www/magento, enter http://www.example.com/magento |
| <>RULE_ID>> | Generate Coupons rule ID. Get this value by clicking Promotions > Shopping Cart Price Rules . |

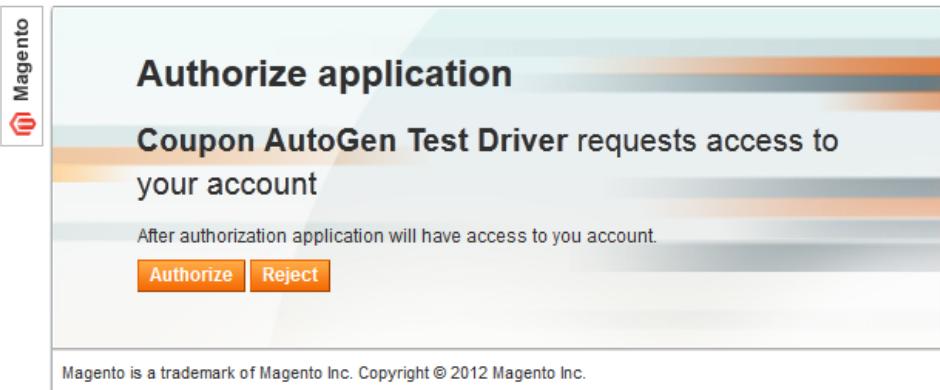
- Save the file and close the text editor.

Running the Test Script

To run the test script:

- Start a web browser.
- In the browser's address or location field, enter:

```
http://magento-server-host-or-ip[:port]/rest_test.php
```
- When prompted, enter the login credentials of the OAuth consumer you created as discussed in [Creating an OAuth Consumer for Your Test Script](#).
- Click **Login**.
- When prompted, click **Authorize** to grant authorization for the script to access your OAuth consumer account, as the following figure shows.



- When prompted, log in as that user.

After you log in, two new coupon codes display as follows to confirm you successfully used the API. If your browser displays a page like this one, you've successfully implemented the Coupon AutoGen REST API!

New coupon codes:
 --> F3H0PJN
 --> 9SZ94OQ

To optionally see these codes in the Admin Panel:

- Log in to the Admin Panel as an administrator.
- Click **Promotions > Shopping Cart Price Rules**.
- Click **Generate Coupons**.
- In the left navigation bar, click **Manage Coupon Codes**.

The codes you generated manually earlier display with the new codes (highlighted in red) as the following figure shows.

| Coupon Code | Created On | Used | Times Used |
|--------------------|-------------------------|------|------------|
| TEST-3NPAKMSA2NPD- | Dec 10, 2012 3:33:36 AM | No | 0 |
| TEST-07ZH7B5NB809- | Dec 10, 2012 3:33:36 AM | No | 0 |
| TEST-SXFK94N4UOD- | Dec 10, 2012 3:33:36 AM | No | 0 |
| 9SZ94OQ | Dec 11, 2012 3:39:07 AM | No | 0 |
| F3H0PJN | Dec 11, 2012 3:39:07 AM | No | 0 |

Re-Enabling Magento's Cache

Only after successfully completing the test, you should re-enable Magento's caching system, so performance returns to normal.

To re-enable the Magento cache:

- In the Magento Admin Panel, click **System > Cache Management**.
- On the Cache Storage Management page, click **Select All** (in the upper-left of the page). A check mark displays next to each Magento cache in the list.
- Click **Enable** from the **Actions** list.
- Click **Submit**.
 Magento enables the selected caches. You can tell because the red DISABLED status indicator is replaced by a green ENABLED indicator.
- Click **Select All** again.
- Click **Refresh** from the **Actions** list.
- Click **Submit**.
 Magento reloads the selected caches. Magento now performs much faster.

Congratulations! You have successfully added the Coupon AutoGen API to Magento's REST API. Following the same procedure, you can expose lots of Magento functionality to external programs.

Troubleshooting Suggestions

The following sections discuss solutions to issues you might encounter when setting up this demonstration:

- [OAuth Package Installation Error: 'make' failed](#)
- [CouponDemo API Calls Options Don't Display in the Admin Panel](#)
- [Errors Running rest_test.php](#)

OAuth Package Installation Error: 'make' failed

Problem: OAuth package installation fails with the error ERROR: `make` failed.

Description: In some cases, the `pecl install oauth` command does not install a C compiler. If you encounter the following error, you must install the `make` package; otherwise, OAuth won't compile:

```
1: make: not found
ERROR: `make' failed
```

Solution:

1. Enter the following commands in the order shown as a user with `root` privileges:

```
apt-get install make
pecl install oauth
```

2. Make sure the message `Build process completed successfully` displays to indicate OAuth compiled successfully.

If the following displays, you must edit your `php.ini` file to find the OAuth library:

```
configuration option "php_ini" is not set to php.ini location
You should add "extension=oauth.so" to php.ini
```

3. Open `php.ini` in a text editor.

If you're not sure where it's located look in the `phpinfo.php` page output. Add the following anywhere in `php.ini`:

```
[OAuth]
extension=oauth.so
```

4. Save your changes to `php.ini` and exit the text editor.

5. Enter the following command to restart the Apache web server.

```
service apache2 restart
```

6. Verify the OAuth installation succeeded as discussed in [Confirming that OAuth Installed Successfully](#).

CouponDemo API Calls Options Don't Display in the Admin Panel

Problem: After setting up the CouponDemo configuration files, the **CouponDemo API Calls**

checkboxes do not display in the Admin Panel. A sample is shown [in a figure earlier in this guide](#).

Description: The **CouponDemo API Calls** checkboxes display to indicate you set up the module correctly. If they don't display, either the Magento cache hasn't been entirely cleared or there's something wrong with the directory structure or configuration files.

Solution: Use the following steps to isolate and correct the issue:

1. Make sure your directory structure is set up *exactly* as shown in [this figure earlier in this guide](#).
2. Make sure you copied the *exact text* from the sample configuration files discussed in [Creating Configuration Files](#). *Do not change anything*, and remember that case is important.
3. Clear the Magento cache in all of the following ways:
 - As a user with sufficient privileges to delete files and directories in the Magento installation, enter the following commands:

```
rm -r magento-install-dir/var/cache
rm -r magento-install-dir/var/session
```
 - Log in to the Magento Admin Panel as an administrator and click **System > Cache Management**.
Click **Flush Magento Cache**.
Click **Flush Cache Storage** and follow the prompts on your screen to delete the cache storage.
4. Log out of the Magento Admin Panel and log back in.
5. Click **System > REST - Roles**.
6. Click **Coupon Auto Generate Demo**.
7. In the left navigation bar, click **Role API Resources**.
8. If the **CouponDemo API Calls** checkboxes do not display, double-check [all of the .xml configuration files](#) to make sure there is *no leading white space* (that is, there are no blank lines at the beginning of the files).
Remove any blank lines and save the .xml files.

Errors Running rest_test.php

The following sections discuss issues you might encounter when you run `rest_test.php` in a web browser:

- [Problem: Invalid auth/bad request](#)
- [Invalid auth/bad request: Rule was not found](#)
- [Invalid auth/bad request: /magento/oauth/initiate was not found on this server](#)

Invalid auth/bad request:

`oauth_problem=consumer_key_rejected`

The following error displays in the web browser:

```
Invalid auth/bad request (got a 401, expected HTTP/1.1 20X or a redirect)
{"messages": {"error": [{"code": 401, "message": "oauth_problem=consumer_key_rejected"}]}}
```

Description: Your OAuth authentication attempt failed because the credentials are incorrect.

Solution: Open `rest_test.php` in a text editor and verify the values of the following:

```
$consumerKey = 'value';
$consumerSecret = 'value';
```

You can find these values in the `oauth_consumer` database table or in the Admin Panel: **System > Web Services > REST - OAuth Consumers**.

After verifying the correct values, save your changes to `rest_test.php` and try again.

Invalid auth/bad request: Rule was not found

The following error displays in the web browser:

```
Invalid auth/bad request (got a 404, expected HTTP/1.1 20X or a redirect)
{"messages": [{"error": [{"code": 404, "message": "Rule was not found."}]}]}
```

Description: The shopping cart promotion rule could not be found.

Solution: Open `rest_test.php` in a text editor and verify the value of the following:

```
$ruleId = value;
```

You can find this value in the Admin Panel: **Promotions > Shopping Cart Price Rules**.

Change the value in `rest_test.php`, save it, and try again.

Invalid auth/bad request: /magento/oauth/initiate was not found on this server

The following error displays in the web browser:

```
Invalid auth/bad request (got a 404, expected HTTP/1.1 20X or a redirect)
Not Found
The requested URL /magento/oauth/initiate was not found on this server.
```

Description: The HTTP redirect failed, most likely because web server rewrites are not properly enabled.

Solution: Make sure web server rewrites are enabled. The procedure you use depends on your web server and operating system. An example for Ubuntu can be found [here](#).

Next Steps

Refer to [Magento APIs—REST](#) for documentation explaining how the Magento's REST API framework works.

2 months ago

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Magento for Developers: Part 1—Introduction to Magento

by [Alan Storm](#), updated for Magento 1.12

[Edit this page on GitHub](#) 

Other articles in this series:

- [Part 1—Introduction to Magento](#)
- [Part 2—The Magento Config](#)
- [Part 3—Magento Controller Dispatch](#)
- [Part 4—Magento Layouts, Blocks and Templates](#)
- [Part 5—Magento Models and ORM Basics](#)
- [Part 6—Magento Setup Resources](#)
- [Part 7—Advanced ORM: Entity Attribute Value](#)
- [Part 8—Varien Data Collections](#)

What is Magento? It's the most powerful online eCommerce platform in the universe and is changing the face of eCommerce forever. :-)

Of course, you already know that. What you may not realize is Magento's also an object-oriented PHP Framework that can be used to develop modern, dynamic web applications that tap into Magento's powerful eCommerce features.

This is the first in a series of articles in which we're going to go on a whirlwind tour of Magento's programming framework features. Don't worry if you don't follow everything immediately. As you study the system more everything in this article will start to make sense, and you'll soon be the envy of your colleagues stuck working with more primitive PHP systems.

In this article...

- [Code Organized in Modules](#)
- [Configuration-Based MVC](#)
- [Controllers](#)
- [Context-Based URI Model Loading](#)
- [Models](#)
- [Helpers](#)
- [Layouts](#)
- [Observers](#)
- [Class Overrides](#)
- [Wrap Up](#)

Or for the more visually oriented [Magento_MVC.pdf](#).

Code Organized in Modules

Magento organizes its code into individual Modules. In a typical PHP **Model-View-Controller (MVC)** application, all the Controllers will be in one folder, all the Models in another, etc. In Magento, files are grouped together based on functionality, which are called **modules** in Magento.

Magento's Code

For example, you'll find Controllers, Models, Helpers, Blocks, etc. related to Magento's checkout functionality in

```
app/code/core/Mage/Checkout
```

You'll find Controllers, Models, Helpers, Blocks, etc. related to Magento's Google Checkout functionality in

```
app/code/core/Mage/GoogleCheckout
```

Your Code

When you want to customize or extend Magento, rather than editing core files directly, or even placing your new Controllers, Models, Helpers, Blocks, etc. next to Magento code, you'll create your own Modules in

```
app/code/local/Package/Modulename
```

Package (also often referred to as a **Namespace**) is a unique name that identifies your company or organization. The intent is that each member of the world-wide Magento community will use their own Package name when creating modules in order to avoid colliding with another user's code.

When you create a new Module, you need to tell Magento about it. This is done by adding an XML file to the folder:

```
app/etc/modules
```

There are two kinds of files in this folder, the first enables an individual Module, and is named in the form:

Packagename_Modulename.xml

The second is a file that will enable multiple Modules from a Package/Namespace, and is named in the form:

Packagename_All.xml. Note it is only used by the core team with the file Mage_All.xml. It is **not recommended** to activate several modules in a single file, as this breaks the modularity of your modules.

Configuration-Based MVC

Magento is a **configuration-based** MVC system. The alternative to this would a **convention-based** MVC system.

In a convention-based MVC system, if you wanted to add, say, a new Controller or maybe a new Model, you'd just create the file/class, and the system would pick it up automatically.

In a configuration-based system, like Magento, in addition to adding the new file/class to the codebase, you often need to explicitly tell the system about the new class, or new group of classes. In Magento, each Module has a file named config.xml. This file contains all the relevant configuration for a Magento Module. At runtime, all these files are loaded into one large configuration tree.

For example, want to use Models in your custom Module? You'll need to add some code to config.xml that tells Magento you want to use Models, as well as what the base class name for all your Models should be.

```
<models>
    <packagename>
        <class>Packagename_Modulename_Model</class>
    </packagename>
</models>
```

The same goes for Helpers, Blocks, Routes for your Controllers, Event Handlers, and more. Almost anytime you want to tap into the power of the Magento system, you'll need to make some change or addition to your config file.

Controllers

In any PHP system, the main PHP entry point remains a PHP file. Magento is no different, and that file is index.php.

However, you never CODE in index.php. In an MVC system, index.php will contain code/calls to code that does the following:

1. Examines the URL
2. Based on some set of rules, turns this URL into a Controller class and an Action method (called Routing)

3. Instantiates the Controller class and calls the Action method (called dispatching)

This means the **practical** entry point in Magento (or any MVC-based system) is a method in a Controller file. Consider the following URL:

```
http://example.com/catalog/category/view/id/25
```

Each portion of the path after the server name is parsed as follows.

Front Name - catalog

The first portion of the URL is called the front name. This, more or less, tells magento which Module it can find a Controller in. In the above example, the front name is *catalog*, which corresponds to the Module located at:

```
app/code/core/Mage/Catalog
```

Controller Name - category

The second portion of the URL tells Magento which Controller it should use. Each Module with Controllers has a special folder named 'controllers' which contains all the Controllers for a module. In the above example, the URL portion category is translated into the Controller file

```
app/code/core/Mage/Catalog/controllers/CategoryController.php
```

Which looks like

```
class Mage_Catalog_CategoryController extends Mage_Core_Controller_Front_Action
{
}
```

All Controllers in the Magento cart application extend from Mage_Core_Controller_Front_Action.

Action Name - view

Third in our URL is the action name. In our example, this is "view". The word "view" is used to create the Action Method. So, in our example, "view" would be turned into "viewAction"

```
class Mage_Catalog_CategoryController extends Mage_Core_Controller_Front_Action
{
    public function viewAction()
    {
        //main entry point
    }
}
```

People familiar with the Zend Framework will recognize the naming convention here.

Parameter/Value - id/25

Any path portions after the action name will be considered key/value GET request variables. So, in our example, the "id/25" means there will get a GET variable named "id", with a value of "25".

As previously mentioned, if you want your Module to use Controllers, you'll need to configure them. Below is the configuration chunk that enables Controllers for the Catalog Module

```
<frontend>
    <routers>
        <catalog>
            <use>standard</use>
            <args>
                <module>Mage_Catalog</module>
                <frontName>catalog</frontName>
            </args>
        </catalog>
    </routers>
</frontend>
```

Don't worry too much about the specifics right now, but notice the

```
<frontName>catalog</frontName>
```

This is what links a Module with a URL frontname. Most Magento core Modules choose a frontname that is the same as their Module name, but this is not required.

Multiple Routers

The routing described above is for the Magento cart application (often called the frontend). If Magento doesn't find a valid Controller/Action for a URL, it tries again, this time using a second set of Routing rules for the Admin application. If Magento doesn't find a valid **Admin** Controller/Action, it uses a special Controller named `Mage_Cms_IndexController`.

The CMS Controller checks Magento's content Management system to see if there's any content that should be loaded. If it finds some, it loads it, otherwise the user will be presented with a 404 page.

For example, the main magento "index" page is one that uses the CMS Controller, which can often throw newcomers for a loop.

Context-Based URI Model Loading

Now that we're in our Action method entry point, we'll want to start instantiating classes that do things. Magento offers a special way to instantiate Models, Helpers and Blocks using static factory methods on the global Mage class. For example:

```
Mage::getModel('catalog/product');
Mage::helper('catalog/product');
```

The string 'catalog/product' is called a Grouped Class Name. It's also often called a URI. The first portion of any Grouped Class Name (in this case, catalog), is used to lookup which Module the class resides in. The second portion ('product' above) is used to determine which class should be loaded.

So, in both of the examples above, 'catalog' resolves to the Module `app/code/core/Mage/Catalog`.

Meaning our class name will start with `Mage_Catalog`.

Then, product is added to get the final class name

```
Mage::getModel('catalog/product');
Mage_Catalog_Model_Product

Mage::helper('catalog/product');
Mage_Catalog_Helper_Product
```

These rules are bound by what's been setup in each Module's config file. When you create your own custom Module, you'll have your own grouped classnames (also called classgroups) to work with `Mage::getModel('myspecialprefix/modelname')`.

You don't **have** to use Grouped Class Names to instantiate your classes. However, as we'll learn later, there are certain advantages to doing so.

Magento Models

Magento, like most frameworks these days, offers an Object Relational Mapping (ORM) system. ORMs get you out of the business of writing SQL and allow you to manipulate a datastore purely through PHP code. For example:

```
$model = Mage::getModel('catalog/product')->load(27);
$price = $model->getPrice();
$price += 5;
$model->setPrice($price)->setSku('SK83293432');
$model->save();
```

In the above example we're calling the methods "getPrice" and "setPrice" on our Product. However, the `Mage_Catalog_Model_Product` class has no methods with these names. That's because Magento's ORM uses PHP's magic `__call` method to implement getters and setters.

Calling the method `$product->getPrice()` will "get" the Model attribute "price".

Calling `$product->setPrice()` will "set" the Model attribute "price". All of this assumes the Model class doesn't already have methods named `getPrice` or `setPrice`. If it does, the magic methods will be bypassed. If you're interested in the implementation of this, checkout the `Varien_Object` class, which all Models inherit from.

If you wanted to get all the available data on a Model, call `$product->getData()` to get an array of all the attributes.

You'll also notice it's possible to chain together several calls to the set method:

```
$model->setPrice ($price)->setSku ('SK83293432');
That's because each set method returns an instance of the Model. This is a pattern you'll see used in much of the Magento codebase.
```

Magento's ORM also contains a way to query for multiple Objects via a Collections interface. The following would get us a collection of all products that cost \$5.00

```
$products_collection = Mage::getModel('catalog/product')
->getCollection()
->addAttributeToSelect('*')
->addFieldToFilter('price','5.00');
```

Again, you'll notice Magento's implemented a chaining interface here. Collections use the PHP Standard Library to implement Objects that have array like properties.

```

foreach($products_collection as $product)
{
    echo $product->getName();
}

```

You may be wondering what the "addAttributeToSelect" method is for. Magento has two broad types of Model objects. One is a traditional "One Object, One Table" Active Record style Model. When you instantiate these Models, all attributes are automatically selected.

The second type of Model is an Entity Attribute Value (EAV) Model. EAV Models spread data across several different tables in the database. This gives the Magento system the flexibility to offer its flexible product attribute system without having to do a schema change each time you add an attribute. When creating a collection of EAV objects, Magento is conservative in the number of columns it will query for, so you can use addAttributeToSelect to get the columns you want, or addAttributeToSelect('*') to get all columns.

Helpers

Magento's Helper classes contain utility methods that will allow you to perform common tasks on objects and variables. For example:

```
> $helper = Mage::helper('catalog');
```

You'll notice we've left off the second part of the grouped class name. Each Module has a default Data Helper class. The following is equivalent to the above:

```
$helper = Mage::helper('catalog/data');
```

Most Helpers inherit from Mage_Core_Helper_Abstract, which gives you several useful methods by default.

```

$translated_output = $helper->__(Mage::getStoreConfig('catalog/translate'));
if($helper->isModuleOutputEnabled()): //is output for this module on or off?

```

Layouts

So, we've seen Controllers, Models, and Helpers. In a typical PHP MVC system, after we've manipulated our Models we would

1. Set some variables for our view
2. The system would load a default "outer" HTML layout>
3. The system would then load our view inside that outer layout

However, if you look at a typical Magento Controller action, you don't see any of this:

```

/**
 * View product gallery action
 */
public function galleryAction()
{
    if (!$this->_initProduct()) {
        if (isset($_GET['store'])) && !$this->getResponse()->isRedirect()) {
            $this->_redirect('');
        } elseif (!$this->getResponse()->isRedirect()) {
            $this->_forward('noRoute');
        }
        return;
    }
    $this->loadLayout();
    $this->renderLayout();
}

```

Instead, the Controller action ends with two calls

```

$this->loadLayout();
$this->renderLayout();

```

So, the "V" in Magento's MVC already differs from what you're probably used to, in that you need to explicitly kick off rendering the layout.

The layout itself also differs. A Magento Layout is an object that contains a nested/tree collection of "Block" objects. Each Block object will render a specific bit of HTML. Block objects do this through a combination of PHP code, and including PHP .phtml template files.

Blocks objects are meant to interact with the Magento system to retrieve data from Models, while the phtml template files will produce the HTML needed for a page.

For example, the page header Block app/code/core/Mage/Page/Block/Html/Head.php uses the head.phtml file page/html/head.phtml.

Another way of thinking about it is the Block classes are almost like little mini-controllers, and the .phtml files are the view.

By default, when you call

```

$this->loadLayout();

```

```
$this->renderLayout();
```

Magento will load up a Layout with a skeleton site structure. There will be Structure Blocks to give you your `html`, `head`, and `body`, as well as HTML to setup single or multiple columns of Layout. Additionally, there will be a few Content Blocks for the navigation, default welcome message, etc.

"Structure" and "Content" are arbitrary designations in the Layout system. A Block doesn't programmatically know if it's Structure or Content, but it's useful to think of a Block as one or the other.

To add Content to this Layout you need to tell the Magento system something like

```
"Hey, Magento, add these additional Blocks under the "content" Block of the skeleton"
```

or

```
"Hey, Magento, add these additional Blocks under the "left column" Block of the skeleton"
```

This can be done programmatically in a Controller action

```
public function indexAction()
{
    $this->loadLayout();
    $block = $this->getLayout()->createBlock('adminhtml/system_account_edit')
    $this->getLayout()->getBlock('content')->append($block);
    $this->renderLayout();
}
```

but more commonly (at least in the frontend cart application), is use of the XML Layout system.

The Layout XML files in a theme allow you, on a per Controller basis, to remove Blocks that would normally be rendered, or add Blocks to that default skeleton areas. For example, consider this Layout XML file:

```
<catalog_category_default>
    <reference name="left">
        <block type="catalog/navigation" name="catalog.leftnav" after="currency" template="catalog/navigation/left.phtml"/>
    </reference>
</catalog_category_default>
```

It's saying in the catalog Module, in the category Controller, and the default Action, insert the catalog/navigation Block into the "left" structure Block, using the catalog/navigation/left.phtml template.

One last important thing about Blocks. You'll often see code in templates that looks like this:

```
$this->getChildHtml('order_items')
```

This is how a Block renders a nested Block. However, a Block can only render a child Block if the child Block is included as a nested Block in the Layout XML file. In the example above our catalog/navigation Block has no nested Blocks. This means any call to `$this->getChildHtml()` in left.phtml will render as blank.

If, however, we had something like:

```
<catalog_category_default>
    <reference name="left">
        <block type="catalog/navigation" name="catalog.leftnav" after="currency" template="catalog/navigation/left.phtml">
            <block type="core/template" name="foobar" template="foo/baz/bar.phtml"/>
        </block>
    </reference>
</catalog_category_default>
```

From the catalog/navigation Block, we'd be able to call

```
$this->getChildHtml('foobar');
```

Observers

Like any good object-oriented system, Magento implements an Event/Observer pattern for end users to hook into. As certain actions happen during a Page request (a Model is saved, a user logs in, etc.), Magento will issue an event signal.

When creating your own Modules, you can "listen" for these events. Say you wanted to get an email every time a certain customer logged into the store. You could listen for the "customer_login" event (setup in config.xml)

```
<events>
    <customer_login>
        <observers>
            <unique_name>
```

```

<type>singleton</type>
<class>mymodule/observer</class>
<method>iSpyWithMyLittleEye</method>
</unique_name>
</observers>
</customer_login>
</events>

```

and then write some code that would run whenever a user logged in:

```

class Packagename_Mymodule_Model_Observer
{
    public function iSpyWithMyLittleEye($observer)
    {
        $data = $observer->getData();
        //code to check observer data for our user,
        //and take some action goes here
    }
}

```

Class Overrides

Finally, the Magento System offers you the ability to replace Model, Helper and Block classes from the core modules with your own. This is a feature that's similar to "Duck Typing" or "Monkey Patching" in a language like Ruby or Python.

Here's an example to help you understand. The Model class for a product is `Mage_Catalog_Model_Product`.

Whenever the following code is called, a `Mage_Catalog_Model_Product` object is created

```
$product = Mage::getModel('catalog/product');
```

This is a factory pattern.

What Magento's class override system does is allow you to tell the system

```
"Hey, whenever anyone asks for a catalog/product, instead of giving them a Mage_Catalog_Model_Product,
give them a Packagename_Modulename_Model_Foobazproduct instead".
```

Then, if you want, your `Packagename_Modulename_Model_Foobazproduct` class can extend the original product class

```

class Packagename_Modulename_Model_Foobazproduct extends Mage_Catalog_Model_Product
{
}

```

Which will allow you to change the behavior of any method on the class, but keep the functionality of the existing methods.

```

class Packagename_Modulename_Model_Foobazproduct extends Mage_Catalog_Model_Product
{
    public function validate()
    {
        //add custom validation functionality here
        return $this;
    }
}

```

As you might expect, this overriding (or rewriting) is done in the config.xml file.

```

<models>
    <!-- does the override for catalog/product-->
    <catalog>
        <rewrite>
            <product>Packagename_Modulename_Model_Foobazproduct</product>
        </rewrite>
    </catalog>
</models>

```

One thing that's important to note here. Individual classes in **your** Module are overriding individual classes in **other** Modules. You are not, however, overriding the entire Module. This allows you to change specific method behavior without having to worry what the rest of the Module is doing.

Wrap Up

We hope you've enjoyed this whirlwind tour of some of the features the Magento eCommerce system offers to developers. It can be a little overwhelming at first, especially if this is your first experience with a modern, object-oriented PHP system. If you start to get frustrated, take a deep breath, remind yourself that this is new, and new things are hard, but at the end of the day it's just a different way of coding. Once you get over the learning curve you'll find yourself loath to return to other, less powerful systems.

 22 days ago

Did you find what you were looking for?

Yes No

-
- [Become a Contributor](#)
 - [Glossary](#)
 - [Privacy Policy](#)
 - [Terms of Service](#)
 - [License/Trademark FAQ](#)
 - [Release Notes](#)
 - [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Magento 1.x REST API

Welcome to the Magento 1.x REST API home page.

- [Introduction to the Magento 1.x REST API](#)
- [Authentication](#)
 - [OAuth Configuration](#)
- [Common HTTP Status Codes](#)
- [HTTP Methods](#)
- [GET Filters](#)
- [Permission Settings](#)
 - [Roles Configuration](#)
 - [Attributes Configuration](#)
 - [Attributes Description](#)
- [Resources](#)
 - [Inventory](#)
 - [Sales Orders](#)
 - [Order Addresses](#)
 - [Order Comments](#)
 - [Order Items](#)
 - [Products](#)
 - [Product Categories](#)
 - [Product Images](#)
 - [Product Websites](#)
 - [Customer Addresses](#)
 - [Customers](#)
 - [Response Formats](#)
 - [Testing REST Resources](#)

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Magento 1.x SOAP API

Welcome to the Magento 1.x SOAP API home page.

- [Introduction to the Magento 1.x SOAP API](#)

- **Catalog**

- [Catalog Category](#)
 - [Assigned Products](#)
 - [Assign Product](#)
 - [Create Category](#)
 - [Current Store](#)
 - [Category Delete](#)
 - [Category Info](#)
 - [Category Level](#)
 - [Category Move](#)
 - [Remove Product](#)
 - [Category Tree](#)
 - [Category Update](#)
 - [Category Product Update](#)

- [Category Attributes](#)
 - [Current Store](#)
 - [Attribute List](#)
 - [Attribute Options](#)

- [Catalog Product](#)
 - [Product Create](#)
 - [Current Product Store](#)
 - [Product Delete](#)
 - [Get Special Price](#)
 - [Product Info](#)
 - [Product List](#)
 - [List of Additional Attributes](#)

-

- [Set Special Price](#)
- [Product Update](#)
- [Catalog Product Attribute](#)
 - [Add Option](#)
 - [Create Attribute](#)
 - [Current Store](#)
 - [Attribute Info](#)
 - [Attribute List](#)
 - [Attribute Options](#)
 - [Attribute Remove](#)
 - [Remove Option](#)
 - [Attribute Types](#)
 - [Attribute Update](#)
- [Catalog Product Attribute Media](#)
 - [Create](#)
 - [Current Store](#)
 - [Media Info](#)
 - [Media List](#)
 - [Media Remove](#)
 - [Media Types](#)
 - [Media Update](#)
- [Product Attribute Set](#)
 - [Attribute Set Add](#)
 - [Attribute Set Remove](#)
 - [Attribute Set Create](#)
 - [Attribute Set Group Add](#)
 - [Attribute Set Group Remove](#)
 - [Attribute Set Group Rename](#)
 - [Attribute Set List](#)
 - [Attribute Set Remove](#)
- [Catalog Product Custom Option](#)
 - [Product Custom Option Add](#)
 - [Product Custom Option Info](#)
 - [Product Custom Option List](#)
 - [Product Custom Option Remove](#)
 - [Product Custom Option Types](#)
 - [Product Custom Option Update](#)
- [Catalog Product Custom Option Value](#)
 - [Product Custom Option Value Add](#)
 - [Product Custom Option Value Info](#)
 - [Product Custom Option Value List](#)
 - [Product Custom Option Value Remove](#)
 - [Product Custom Option Value Update](#)
- [Catalog Product Downloadable Link](#)

- Downloadable Link Add
- Downloadable Link List
- Downloadable Link Remove
- Catalog Product Link
 - Product Link Assign
 - Product Link Attributes
 - Product Link List
 - Product Link Remove
 - Product Link Types
 - Product Link Update
- Catalog Product Tag
 - Product Tag Add
 - Product Tag Info
 - Product Tag List
 - Product Tag Remove
 - Product Tag Update
- Catalog Product Attribute Tier Price
 - Product Attribute Tier Price Info
 - Product Attribute Tier Price Update
- Catalog Product Types
 - Product Type List
- **Catalog Inventory**
 - Inventory Stock Item List
 - Inventory Stock Item Update
- **Checkout**
 - Cart
 - Cart Create
 - Cart Info
 - Cart License
 - Cart Order
 - Cart Totals
 - Cart Coupon
 - Coupon Add
 - Coupon Remove
 - Cart Customer
 - Customer Addresses
 - Customer Set
 - Cart Payment
 - Payment List
 - Payment Method
 - Cart Product
 - Product Add
 -

Product List

- Product Move To Customer Quote
- Product Remove
- Product Update
- Cart Shipping
 - Shipping List
 - Shipping Method

• Create Your Own API

• Customer

- Customer Group
- Customer Create
- Customer Delete
- Customer Info
- Customer List
- Customer Update
- Customer Address
 - Address Create
 - Address Delete
 - Address Info
 - Address List
 - Address Update

• Directory

- Country List
- Region List

• Sales

- Sales Order
 - Add Comment
 - Order Cancel
 - Order Hold
 - Order Info
 - Order List
 - Order Unhold
- Sales Order Credit Memo
 - Add Comment
 - Memo Cancel
 - Memo Create
 - Memo Info
 - Memo List
- Sales Order Invoice
 - Add Comment
 - Invoice Cancel
 - Invoice Capture
 - Invoice Create
 -

- Invoice Info
- Invoice List
- Sales Order Shipment
 - Add Comment
 - Add Track
 - Shipment Create
 - Shipment Get Carriers
 - Shipment Info
 - Shipment List
 - Remove Track
- **Enterprise Customer Balance**
 - Customer Balance
 - Store Credit
 - Credit History
 - Shopping Cart Customer Balance
 - Balance Remove Amount
 - Balance Set Amount
- **Enterprise Gift Card**
 - Cart Gift Card
 - Gift Card Add
 - Gift Card List
 - Gift Card Remove
 - Gift Card Account
 - Account Create
 - Account Info
 - Account List
 - Account Remove
 - Account Update
 - Gift Card Customer
 - Customer Info
 - Customer Redeem
- **Enterprise Gift Message**
 - Set For Quote
 - Set For Quote Item
 - Set For Quote Product
- **Miscellaneous**
 - Magento Info
 - Store Info
 - Store List
- **WS-I Compliance**

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

How to Apply and Revert Magento Patches

Table of Contents

- [Overview](#)
- [Need More Detail?](#)
- [How to Get Magento Patches](#)
- [How to Apply a Magento Patch](#)
- [How to Apply the SUPEE-8788 Patch](#)
- [Listing Patches You Have Installed](#)
- [How to Revert a Magento Patch](#)
- [Troubleshooting](#)

Overview

This article discusses how to apply and revert Magento patches you get in any of the following ways:

- Magento Support
- Magento Enterprise Edition (EE) support portal
- (Magento partners) from the [partner portal](#)

If you don't already have a patch, [contact Magento Support](#).



Note: This article assumes your patch file name ends in `.sh`. If your patch file name ends in `.patch` or something else, [contact Magento Support](#) before proceeding.

Need More Detail?

For more step-by-step details that are provided here, see one of the following:

- [Magento CE patch instructions](#)
- [Magento EE patch instructions](#)

How to Get Magento Patches

Magento Support provides some patches for Magento CE and EE on [magentocommerce.com](#). This section discusses how to get those patches.

If Magento Support provided a patch to you, skip this section and continue with [How to Apply a Magento Patch](#).

See one of the following sections for specific information about Magento CE or EE:

- [Getting Magento CE Patches](#)
- [Getting Magento EE Patches](#)

Getting Magento CE Patches

To get patches for Magento CE:

1. Log in to [magentocommerce.com/download](#).
(Click **My Account** in the upper right corner of the page.)
If you don't have an account, you can register for one; the account is free.
2. In the Magento Community Edition Patches section, locate the patch to install.
3. From the list next to the patch, choose your CE version.
4. Click **Download**.
5. After the patch downloads, continue with [How to Apply a Magento Patch](#).

Getting Magento EE Patches

To get patches for Magento EE:

1. Log in to [magentocommerce.com](#).
(Click **My Account** in the upper right corner of the page.)
2. Click **Downloads** in the left pane.
3. Click **Magento Enterprise Edition** in the right pane.

The following figure shows an example.

The screenshot shows the 'My Account' interface. On the left, there's a sidebar with links like 'Log Out', 'My Products and Services', 'Account Settings', 'Billing History', 'Support' (which has a red box around it and a circled '3'), 'Downloads' (which has a red box around it and a circled '4'), 'My Magento U', 'Community', 'Developers', and 'Shared Access'. The main area is titled 'Downloads' and contains a message about USPS shipping changes. Below the message is a list of categories: 'Magento Enterprise Edition' (which is highlighted with a red box and circled '4'), 'Magento User Guide', 'Merchants - Resources for Getting Started', 'Magento Payment Bridge', 'Magento Professional Edition', 'Magento U', and 'Virtual Machine'.

4. Click **Support Patches**.
5. Locate the patch to download.
6. Click **Download** corresponding to the patch for the version of EE you're using.
7. After the download completes, continue with the next section.

How to Apply a Magento Patch

To apply a Magento patch:

1. Transfer the patch .sh file to your Magento installation root directory.

 **Note:** This article assumes your patch file name ends in .sh. If your patch file name ends in .patch or something else, [contact Magento Support](#) before proceeding.

For example, /var/www/html/magento.

2. Enter the following commands as a user with sufficient privileges to write to Magento files (typically, the web server user or `root`):

```
chmod +x <patch-file-name>.sh  
./<patch-file-name>.sh
```

A message such as the following displays to confirm the patch installed successfully:

```
Patch was applied/reverted successfully.
```

3. To reapply ownership to the files changed by the patch:

- a. Find the web server user: `ps -o "user group command" -C httpd,apache2`

The value in the USER column is the web server username.

Typically, the Apache web server user on CentOS is `apache` and the Apache web server user on Ubuntu is `www-data`.

- b. As a user with `root` privileges, enter the following command from the Magento installation directory:

```
chown -R web-server-user-name .
```

For example, on Ubuntu where Apache usually runs as `www-data`, enter

```
chown -R www-data .
```

4. Perform any other tasks as instructed by Magento Support.

(For example, some patches require you to stop external services, such as the Solr search engine.)

How to Apply the SUPEE-8788 Patch

We released a [security patch](#) in October, 2016 that might cause issues for some users. This section applies to you if any of the following is true:

- You haven't yet applied the SUPEE-8788 patch and your Magento version is earlier than EE 1.14.1.0 or CE 1.9.1.0.
- You applied version 1 of the SUPEE-8788 patch. (The patch name includes `PATCH_SUPEE-8788_<magento_version>_v1`.)
- You previously applied the SUPEE-1533 patch and you want to apply the SUPEE-8788 patch.
- You're applying the SUPEE-8788 patch as part of an upgrade from an earlier Magento version.

We recommend the following:

- If you applied SUPEE-8788 version 1, revert that patch, revert SUPEE-1533 (version restrictions apply), apply SUPEE-3941 (version restrictions apply), then [apply SUPEE-8788 version 2 or later](#).
- If you haven't applied SUPEE-8788, revert SUPEE-1533 (version restrictions apply), apply SUPEE-3941 (version restrictions apply), then [apply SUPEE-8788](#).

Replace SUPEE-8788 version 1 with version 2 or later

To replace SUPEE-8788 version 1 with version 2 or later:

1. Log in to your Magento server.

2. Open `<your Magento install dir>/app/etc/applied.patches.list` in a text editor.

This file lists all currently applied patches.

3. Determine which patches are already applied. Version 1 of SUPEE-8788 includes `PATCH_SUPEE-8788_<magento_version>_v1` in the name.
4. If your Magento version is EE 1.14.1.0 or CE 1.9.1.0, *and* patch SUPEE-1533 is applied, [revert](#) SUPEE-1533.
5. If your Magento version is earlier than EE 1.14.1.0 or CE 1.9.1.0, *and* SUPEE-3941 is not applied, [apply SUPEE-3941](#).
6. [Get](#) version 2 or later of SUPEE-8788.
7. [Apply](#) version 2 or later of SUPEE-8788.
8. *Magento EE 1.14.2 only.* After applying the SUPEE-8788 patch, remove `test_oauth.php` from your Magento base directory.
9. If you upgraded to Magento CE 1.9.3 or Magento EE 1.14.3 after applying the [SUPEE-8788 patch](#), make sure the following files have been deleted:

```
skin/adminhtml/default/default/media/flex.swf  
skin/adminhtml/default/default/media/upload.swf  
skin/adminhtml/default/default/media/uploadersingle.swf
```

If the files are present, delete them to avoid a potential security exploit. As of Magento CE 1.9.0.0 and Magento EE 1.14.0.0, we no longer distribute `.swf` files with the Magento software.

Apply SUPEE-8788

To apply patch SUPEE-8788:

1. Open <your Magento install dir>/app/etc/applied.patches.list in a text editor.

This file lists all currently applied patches.
2. Verify SUPEE-8788 is *not* applied. If it is, and it's version 1, see [Replace SUPEE-8788 version 1 with version 2 or later](#).
3. Verify whether or not patch SUPEE-1533 is applied. If it is, *and* your Magento version is earlier than EE 1.14.1.0 or CE 1.9.1.0, [revert SUPEE-1533](#).
4. If your Magento version is earlier than EE 1.14.1.0 or CE 1.9.1.0, *and* SUPEE-3941 is not applied, [apply SUPEE-3941](#).
5. [Get](#) version 2 or later of SUPEE-8788.
6. [Apply](#) version 2 or later of SUPEE-8788.
7. *Magento EE 1.14.2 only*. After applying the SUPEE-8788 patch, remove `test_oauth.php` from your Magento base directory.
8. If you upgraded to Magento CE 1.9.3 or Magento EE 1.14.3 after applying the [SUPEE-8788 patch](#), make sure the following files have been deleted:

```
skin/adminhtml/default/default/media/flex.swf  
skin/adminhtml/default/default/media/uploader.swf  
skin/adminhtml/default/default/media/uploaderSingle.swf
```

If the files are present, delete them to avoid a potential security exploit. As of Magento CE 1.9.0.0 and Magento EE 1.14.0.0, we no longer distribute .swf files with the Magento software.

Listing Patches You Have Installed

If you're not sure which patches are already applied, open <your Magento install dir>/app/etc/applied.patches.list.

How to Revert a Magento Patch

If applying the patch results in errors, [contact Magento Support](#). If you are instructed to do so, revert the patch:

1. Change to your Magento installation directory.
2. Enter the following command as a user with sufficient privileges to write to Magento files (typically, the web server user or `root`):

```
sh patch-file-name.sh -R
```

Troubleshooting

If you get an error when you run the patch, use the following suggestions:

- Verify the patch is located in your Magento installation root directory.
Ubuntu example: `/var/www/magento`
CentOS example: `/var/www/html/magento`
- Verify you're running the patch with sufficient privileges.
Typically, this means running it as the web server user or as a user with `root` privileges.
- Try running the patch again.
If problems persist, [contact Magento Support](#).

2 months ago

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020.

Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Discover credit card validation issue: Magento EE 1.9.1.1—1.13.1.0 and CE 1.4.2.0—1.8.1.0

Magento has fixed an issue that prevented some Discover credit cards from validating properly. The issue was that certain Discover credit card number ranges were not recognized as being valid. As a result of the fix, all Discover cards should validate properly.

This fix applies to:

- Magento Enterprise Edition (EE) versions 1.9.1.1–1.13.1.0
- Magento Community Edition (CE) versions 1.4.2.0–1.8.1.0



Important: This is *not* a security threat. No data has been compromised or misused. It affects only the ability to validate certain credit card number ranges as valid Discover card numbers.

To get the patch for Magento CE, [submit a Magento CE help request](#).

To get the patch for Magento EE:

1. Log in to www.magentocommerce.com.
2. In the left pane, click **Downloads**.
3. In the right pane, click **Magento Enterprise Edition**.
4. Follow the prompts on your screen to download the SUPEE-2725 patch for your version of EE.
5. Apply the patch as discussed in [How to Apply and Revert Magento Patches](#).



Important: If you applied an earlier version of this patch, you must first revert the old patch.

 2 months ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Patches for Magento Enterprise Edition (EE) Versions 1.10–1.12

We'd like to draw your attention to new patches that were recently posted to the Partner Portal and Support Center. These patches deliver important improvements, such as improving security of the Magento Connect Manager and making it easier to install community-created translation packages.

General Magento Connect Patches

Patch name:

- PATCH_SUPEE-3941_EE_1.10.1.0_v1-2014-08-12-12-13-15.sh for EE 1.10.x
- PATCH_SUPEE-3941_EE_1.11.2.0_v1-2014-08-12-12-11-32.sh for EE 1.11.x
- PATCH_SUPEE-3941_EE_1.14.0.1_v1-2014-08-12-12-10-06.sh for EE 1.12.x

Result of applying this patch:

- When you install a community-created translation package, the translation provided by the package overwrites any existing translations for the same items. This enables you to more easily install packages with translations.
- To improve security, Magento Connect now uses HTTPS by default to download extensions, rather than FTP.
- Extension developers can now create an extensions with a dash character in the name. Merchants can install those extensions without issues.
- Magento administrators who attempt to install an extension with insufficient file system privileges are now informed. Typically, the Magento Admin Panel runs as the web server user. If this user has insufficient privileges to the *your Magento install dir/app/code/community* directory structure, the Magento administrator sees an error message in the Magento Connect Manager. To set file system permissions appropriately, see [After You Install Magento: Recommended File System Ownership and Privileges](#).

How to Get This Patch

To get this patch:

1. Log in to www.magentocommerce.com.
2. In the left pane, click **Downloads**.
3. In the right pane, click **Magento Enterprise Edition**.
4. Follow the prompts on your screen to download a patch for your version of EE.
5. Apply the patch as discussed in [How to Apply and Revert Magento Patches](#).

Solr Search Results and Indexing

Patch name: PATCH_SUPEE-3630_EE_1.12.x_v1.sh for EE 1.12.x. After applying this patch, your search engine returns search results to users on the web store while reindexing is underway. We recommend you install this patch if you're using the Solr search engine. *This patch can result in slower performance if you're using the default MySQL Full Text search engine.*

To get this patch, [contact Magento Support](#).

 2 months ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Error Using Payflow with Magento Enterprise Edition (EE) 1.12.0.2

Problem

A customer applies a coupon code when checking out in a web store that runs EE 1.12.0.2 and is configured to use the Payflow Pro or Payflow Express payment methods. The following error displays:

`PayPal gateway rejected the request. Field format error: 10431-Item amount is invalid`

Solution

To resolve this issue, [contact Magento Support](#) and request the patch for support issue ID SUPEE-1474.

Then apply the patch as discussed in [How to Apply and Revert Magento Patches](#).

 2 months ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020.

Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Getting the PHP 5.4 patch for Magento Enterprise Edition (EE) and Community Edition (CE)

Magento has a patch that enables you to use PHP 5.4.x with Magento Enterprise Edition (EE) and Magento Community Edition (CE).

This patch applies to:

- Magento CE versions 1.6.0.0–1.8.1.0
- Magento EE versions 1.11.0.0–1.13.1.0

To get the patch for Magento CE:

1. Go to www.magentocommerce.com/download.
2. Click **My Account** in the upper right corner of the page.
3. Log in when prompted.
4. Scroll to the Magento Community Edition Patches section.
5. Follow the prompts on your screen to get the PHP 5.4 support patch.

To get the patch for Magento EE:

1. Log in to www.magentocommerce.com.
2. In the left pane, click **Downloads**.
3. In the right pane, click **Magento Enterprise Edition**.
4. Follow the prompts on your screen to download the PHP 5.4 Compatibility patch.
5. Apply the patch as discussed in [How to Apply and Revert Magento Patches](#).

 2 months ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Resolving a Remote Code Execution Exploit

Table of Contents

- [Overview](#)
- [Versions Affected](#)
- [Getting the Patch](#)
- [Determining Your Vulnerability to the File System Attack](#)
- [Resolving the File System Vulnerability](#)

Overview

We recently identified potential exploits that:

- Enable an attacker to execute arbitrary code on your Magento server.
- Create files with a .csv extension, create writable directories, and change the permission of existing files to world-writable (777).



Note: The preceding exploits require the attacker to have administrative access to your Magento Admin Panel Dashboard. You can resolve these issues with the patch discussed in this article.

Creating files with a .csv extension can lead to executing files like `php.csv` (only under circumstances discussed in this article). The ability to run code with a .csv extension is dangerous itself and could be combined with other attacks; for example, targeting other software installed on the server.

You can resolve this issue by changing your server's configuration as discussed in [Resolving the File System Vulnerability](#).

Although Magento code is protected by a hash value, the possibility of a successful exploit cannot be eliminated because of the low entropy of the hash secret value.

We strongly recommend you to take precautions discussed in this article *and* apply a patch for your version of Magento Enterprise Edition or Community Edition.

Versions Affected

Magento software versions affected: The issue affects all shipping versions of Magento Community Edition (CE) and Enterprise Edition (EE).

Operating system versions affected:

- CentOS 5.x and 6.x
- RedHat Enterprise Linux 5.x and 6.x

Getting the Patch

The following table shows the patch you should get for your version of CE or EE.

| VERSION | PATCH |
|----------------------------------|---------------------------------|
| EE 1.13 and 1.14, CE 1.8 and 1.9 | SUPEE-1533_EE_1.13.x_v1.patch |
| EE 1.12, CE 1.7 | SUPEE-1533_EE_1.12.x_v1.patch |
| EE 1.11, CE 1.6 | SUPEE-1533_EE_1.11.x_v1.patch |
| EE 1.10.1, CE 1.5.1 | SUPEE-1533_EE_1.10.1.x_v1.patch |
| EE 1.10.0.1, CE 1.5.0.1 | SUPEE-1533_EE_1.10.0.x_v1.patch |
| EE 1.9 | SUPEE-1533_EE_1.9.x_v1.patch |

To get and apply your patch, see [How to Apply and Revert Magento Patches](#).

 **Important:** After applying your patch, Magento strongly recommends you evaluate your vulnerability and configure PHP as discussed in [Resolving the Vulnerability](#).

Determining Your Vulnerability to the File System Attack

To determine if you're vulnerable to execution of PHP code with a non-PHP extension, search your web server configuration file for the following string:

```
AddHandler application/x-httpd-php .php
```

The Apache configuration file is typically `/etc/httpd/conf/httpd.conf`

To confirm you're vulnerable:

1. Create a file named `test.php.csv` anywhere in your web server's doocroot with the following contents:

```
<?php  
phpinfo()
```

2. In a web browser, display that page. (For example, `http://www.example.com/path/test.php.csv`)
3. If your browser saves the file or prompts you to save the file instead of displaying it, your server is *not* vulnerable. You can ignore the rest of this article.
4. If a page similar to the following displays, your server *is* vulnerable. Continue with the next section.



Resolving the File System Vulnerability

 **Note:** Magento strongly recommends you perform all tasks discussed in this section in a development or testing environment and *not* in a production environment.

To resolve this vulnerability, you must log in to the Magento server as a user with `root` privileges or as a user with permissions to change the web server configuration.

To resolve the vulnerability:

1. Comment out the directive in `httpd.conf` by preceding it with a pound sign (#) as follows:

```
# AddHandler application/x-httpd-php .php
```

2. Add a block similar to the following:

```
<FilesMatch \.php$>
SetHandler application/x-httpd-php
</FilesMatch>
```

Some operating systems, like Red Hat Enterprise, might require different syntax such as:

```
<FilesMatch \.php$>
    SetHandler php5-script
</FilesMatch>
```

The regular expression in this setting matches .php only to the final extension in the file name, applying the handler only to PHP files and preventing PHP from executing.

2 months ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

How to Install and Configure the Solr Search Engine With Magento Enterprise Edition (EE) 1.x

Contents

- [Overview](#)
- [Simple Comparison of Solr and MySQL Search Engines](#)
- [Prerequisites](#)
- [Installing Prerequisite Software](#)
- [Basic Solr Configuration](#)
- [Reindexing Catalog Search and Refreshing the Full Page Cache](#)
- [Scripting Solr Startup and Shutdown](#)
- [Next Steps](#)

Overview

Magento Enterprise Edition (EE) versions 1.8.0.0 and later enable you to configure either of the following as a catalog search engine:

- Full text search using the MySQL database (the default)
- The [Apache Solr search engine](#)

Solr enables you to provide your web store users with a powerful full-text search engine that includes:

- [Faceted search](#)
- [Dynamic clustering using SolrCloud](#)
- [Database integration](#)

See one of the following sections for more information:

- [Intended Audience and Purpose of this Guide](#)
- [Comparing the Search Options](#)
- [Support Matrix for Solr and Magento EE](#)
- [More Information About the Solr Solution](#)

Intended Audience and Purpose of this Guide

This guide is intended for Magento EE administrators and systems integrators who have some familiarity with search engines—ideally, who also have Solr configuration experience. No programming is required to perform the tasks discussed in this guide.

This guide discusses a simple Solr configuration that uses the example Solr configuration provided with Solr, default Solr integration options provided with Magento EE, and also explains how to configure Magento EE to use Solr. Advanced configuration tasks—such as setting up dictionaries—are beyond the scope of this guide.



Note: The example Solr configuration is *not* intended to be used in a production site. It's for testing and development only. Because it's simple to use which, it's a great way for you to learn more about Solr.

Comparing the Search Options

The following table provides a quick comparison between Magento with the default MySQL full text search and Magento with Solr search.

| Feature | Magento with MySQL full-text search | Magento with Solr search |
|---|---|--------------------------|
| Full text search | Yes and also supports two additional search modes: <ul style="list-style-type: none">● Like● Combined (like and full text) | Yes† |
| Search recommendations | Yes | Yes |
| Faceted search (used in layered navigation) | Yes | Yes |
| Range (such as price range) | Yes | Yes |
| Sort-by options (for example, sort by relevance) | Yes | Yes |
| Zero results tips or results correction | No | Yes |
| Suggestions | No | Yes |
| Clustering | No | Yes |
| Attribute weight based on attribute settings | No | Yes |
| Search localized characters | No | Yes |
| Word delimiter (for example, searching for <code>spider man</code> or <code>spiderman</code> return <code>spider-man</code>) | No | Yes |

†—"Like" searching is supported by MySQL full text search but not by Solr. Defined by the `Mage_CatalogSearch_Model_Resource_Fulltext::prepareResult()` class, like searching joins each term in your search using LIKE statements combined by OR. Like searching is best used in stores that have simple products where users search for specific terms.

Support Matrix for Solr and Magento EE

The following table summarizes what versions of Magento EE work with what versions of Solr.

| Magento EE version | Compatible Solr version(s) | Comment |
|--------------------|---|--|
| EE 1.14.x.x | Any version up to 3.6.2 . | No patch is required to use EE 1.14.x.x with Solr 3.6.2. Click here to browse the list of all available Solr versions. |

| | | |
|------------------------|---|---|
| EE 1.13.x.x | Any version up to 3.6.2 . | Click here to browse the list of all available Solr versions. |
| EE 1.12.0.x | Any version up to 3.5.0 | If you use a later version, such as 3.6.x, see Information About Enterprise Edition (EE) Patches for Apache Solr to get a patch. Click here to browse the list of all available Solr versions. |
| EE 1.8.0.0 to 1.11.0.2 | Any version up to 3.4.0 . | To use a Solr version later than 1.4.x, see Information About Enterprise Edition (EE) Patches for Apache Solr to get a patch. Click here to browse the list of all available Solr versions. |



Note: Magento EE does not support Solr 4.x.

More Information About the Solr Solution

Solr runs as a standalone full-text search server in a servlet container such as Jetty (which is used by the Solr example configuration) and Tomcat.

Solr uses the Lucene Java search library for full-text indexing and search. Your applications interact with Solr using [HTTP POST](#) (in JSON, XML, CSV, or binary formats) to index documents and using [HTTP GET](#) to retrieve search results back as JSON, XML, or a variety of other formats (Python, Ruby, PHP, CSV, binary, and so on). If you're a programmer, try the [Solr tutorial](#). Whether you're a programmer or not, read the [Solr FAQ](#).

No programming is required to implement Solr as discussed in this guide.

Solr's powerful external configuration allows it to be tailored to almost any type of application without Java coding, and it has an extensive plug-in architecture when more advanced customization is required. Solr is highly scalable, providing distributed search and index replication.



Important: Customize the Solr search engine at your own risk. Magento supports only the options displayed in the Admin Panel. Customizing the Solr engine itself, while potentially useful, can cause issues with Magento. If you encounter problems with your customizations, do not contact Magento Support; instead, consult the resources available from the [Apache Solr Wiki](#).

In this guide, you'll use the example configuration provided with Solr and Magento's provided Solr configuration to implement a simple, quick integration with Solr.

Some reasons to use Solr with Magento include:

- Magento ships with a sample Solr configuration that enables you to provide users with a powerful search engine without your needing to customize any code.
- You get better performance of search, catalog views, and [layered navigation](#).
- When the system is under load, Solr avoids frequent updates of the MySQL `catalogsearch_fulltext` table and alleviates issues with database table locks.

Simple Comparison of Solr and MySQL Search Engines

Following is a simple comparison of the default MySQL full-text search and Solr search using Magento EE 1.14.0.0 and Solr 3.6.2. Magento EE catalog content is provided by sample data you can download from Magento.

Among the many options Solr gives you is the option to *suggest* names of products in the event the user enters an incomplete or incorrect search term in your Magento store's **Search** field.

Default MySQL Full-Text Search Using an Incorrect Search Term

Using the default MySQL full-text search, if a user enters an incorrect search term (such as shirrt instead of shirt, no results display as the following figure shows.

The screenshot shows the Madison Island website's search results page for the query 'SHIRRT'. The search bar at the top contains 'shirt'. The main content area displays a message: 'Your search returns no results.' To the right, there is a 'COMPARE PRODUCTS' section stating 'You have no items to compare.' Below the search bar is a newsletter sign-up form with fields for email and a 'SUBSCRIBE' button. At the bottom of the page, there are links for Company (About Us, Contact Us, Customer Service, Privacy Policy), Quick Links (Site Map, Search Terms, Advanced Search), Account (My Account, Orders and Returns), and Connect With Us (Facebook, Twitter, YouTube, Pinterest, RSS). A copyright notice at the very bottom reads '© 2013 Madison Island. All Rights Reserved.'

Solr Search Using an Incorrect Search Term

Using Solr, if a user enters an incorrect search term, suggestions display as the following figure shows.

This screenshot shows the same search results page as above, but using Solr. The search bar still contains 'shirt'. The main content area now includes a red-bordered box containing the message 'Your search returns no results.' followed by 'Did you mean: shirt, shirts?'. The rest of the page structure is identical to the MySQL version, including the newsletter sign-up form, footer links, and copyright notice.

In addition, if a user enters an incomplete search term, Magento provides dictionary-based suggestions as the following figure shows.



SEARCH RESULTS FOR 'COMPU'

Your search returns no results.
Did you mean: [compact](#), [comite](#)

COMPARE PRODUCTS

You have no items to compare.

NEWSLETTER

 [SUBSCRIBE](#)

COMPANY

[ABOUT US](#)
[CONTACT US](#)
[CUSTOMER SERVICE](#)
[PRIVACY POLICY](#)

QUICK LINKS

[SITE MAP](#)
[SEARCH TERMS](#)
[ADVANCED SEARCH](#)

ACCOUNT

[MY ACCOUNT](#)
[ORDERS AND RETURNS](#)

CONNECT WITH US

[FACEBOOK](#)
 [TWITTER](#)
 [YOUTUBE](#)
 [PINTEREST](#)
 [RSS](#)

© 2013 Madison Island. All Rights Reserved.

Prerequisites

The tasks discussed in this guide require the following:

- Magento EE and Solr versions discussed in [Support Matrix for Solr and Magento EE](#).
- Java version 1.6 or later.

To determine if Java is already installed, enter the following command:

```
java -version
```

If the message `java: command not found` displays, you must install the Java SDK as discussed in the next section. If Java is installed, make sure it's version 1.6 or later.

- Tomcat or Jetty servlet container. This guide discusses using Jetty, which comes with Solr. Consult another resource, such as the [Solr Wiki](#), to use Tomcat with Solr.

To see if you're currently running Jetty and to check the version, see [How to find out the version of Jetty](#).

Installing Prerequisite Software

The following sections discuss how to install the prerequisite software:

- [Installing Java](#)
- [Installing Solr 3.6.2 and Jetty](#)

Installing the Java 6 SDK

To install the Java 6 SDK, enter the following command as a user with `root` privileges:

```
apt-get install openjdk-6-jdk
```

To install Java 7, enter the following command as a user with `root` privileges:

```
apt-get install openjdk-7-jdk
```



Note: Java version 7 might not be available for all operating systems. For example, you can search the list of available packages for Ubuntu [here](#).

Installing Solr 3.6.2 and Jetty

The Apache Solr package installs both Solr and Jetty. If Jetty is already installed, see the [Solr with Jetty Wiki](#) for more information.



Note: Tomcat is also a supported servlet container for Solr but discussing how to set up Tomcat with Solr is beyond the scope of this guide. For more information, see the [Solr With Tomcat Wiki](#).

To install Solr and Jetty:

1. Find a suitable Solr distribution.

- [Solr 3.6.2](#)
- [Solr 3.5.0](#)
- [Solr 3.4.0](#)

If you're not sure what version of Solr to use, see [Support Matrix for Solr and Magento EE](#).

2. As a user with root privileges, use wget or a similar command to download the file to an empty directory such as /etc/solr. An example follows.

```
cd empty-directory
wget http://archive.apache.org/dist/lucene/solr/3.6.2/apache-solr-3.6.2.tgz
```

Messages similar to the following display to confirm a successful download.

```
wget http://archive.apache.org/dist/lucene/solr/3.6.2/apache-solr-3.6.2.tgz
--2012-11-01 16:20:16--  http://archive.apache.org/dist/lucene/solr/3.6.2/apache-solr-3.6.2.tgz
Resolving archive.apache.org... 140.211.11.131
Connecting to archive.apache.org|140.211.11.131|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 78673382 (75M) [application/x-tar]
Saving to: `apache-solr-3.6.2.tgz'

100%[=====] 78,673,382 1.41M/s   in 36s

2012-11-01 16:20:52 (2.08 MB/s) - `apache-solr-3.6.2.tgz' saved [78673382/78673382]
```

3. Unpack the Solr installation; an example follows.

```
tar -xvf apache-solr-3.6.2.tgz
```

4. Continue with the next section.

Configuring Solr to Work With Magento

The following topics discuss how to configure Solr to work with Magento EE:

- [Copying the Magento Solr Configuration and Starting Solr](#)
- [Configuring Magento to Work With Solr](#)

Copying the Magento Solr Configuration and Starting Solr

Magento comes packaged with a sample Solr configuration you can use and customize. To get started, you'll copy the Magento configuration to Solr, replacing any existing files. After that you can start Solr and begin configuring Magento to work with it.



Note: The example Solr configuration is *not* intended to be used in a production site. It's for testing and development only. It's simple to use which makes it a great way for you to learn more about Solr.

To copy the Magento Solr configuration:

- As a user with `root` privileges, enter the following commands in the order shown to copy over the Solr configuration with the one packaged with Magento EE:

```
cd [your Solr install dir]/example/solr/conf  
cp -R [your Magento install dir]/lib/Apache/Solr/conf/* .
```

For example, if Solr is installed in `/etc/solr/apache-solr-3.6.2` and Magento is installed in `/var/www/magento`, enter:

```
cd /etc/solr/apache-solr-3.6.2/example/solr/conf  
cp -R /var/www/magento/lib/Apache/Solr/conf/* .
```



Note: If you're prompted to overwrite files, try the command `\cp -R [your Magento install dir]/lib/Apache/Solr/conf/* .`

- CentOS with Tomcat 6 only.* If you're using Tomcat 6 on CentOS, you must modify `[your Solr install dir]/example/solr/conf/solrconfig.xml`

Locate the following line:

```
<dataDir>${solr.data.dir:./solr/data}</dataDir>
```

Change it to:

```
<dataDir>${solr.data.dir:}</dataDir>
```

- As a user with `root` privileges, enter the following command to start Solr:

```
java -jar [your Solr install dir]/example/start.jar
```



Note: This method for starting Solr is for convenience and testing purposes only. In a production environment, you should start and stop Solr using a script as discussed in [Scripting Solr Startup and Shutdown](#).

Configuring Magento to Work With Solr

This section discusses how to configure Magento EE to use the Solr search engine.

To configure Magento to work with Solr:

- Log in to the Magento Admin Panel as an administrator.
- Click **System > Configuration > CATALOG > Catalog**.
- In the right pane, expand **Catalog Search**.
- The following table shows the minimum amount of information to enter to test the connection to your Solr search engine.

Leave all other values at their defaults.

| Option | Description |
|----------------------|---|
| Search Engine | Click Solr |
| Solr Server Hostname | Enter the fully qualified hostname or IP address of the machine running Solr. (If Solr is running on the same host as Magento, you can optionally use 127.0.0.1.) |
| Solr Server Port | Enter Solr's listen port. (The example Jetty servlet container uses 8983. The default for Tomcat is usually 8080.) |
| Solr Server Username | <i>Optional.</i> Enter a username for Solr authentication , if desired. |
| Solr Server Password | <i>Optional.</i> Enter the user's password, if desired. |
| Solr Server Timeout | Enter a connection timeout value, in seconds. |
| Solr Server Path | Specifies the path and name of the Solr web application. The path used by the example Solr configuration is <code>solr</code> . If you customized Solr, the value you enter in this field must exactly match the value of <code>webapp_name=value</code> in <code>[your Solr install dir]/example/solr/conf/scripts.conf</code> . |
| Indexation Mode | <p>Specifies how Solr processes indexed content.</p> <p>From the Indexation Mode list, click one of the following:</p> <ul style="list-style-type: none">● Final commit (Default, recommended): After you reindex the content search index, Solr starts processing content. Users see results from content that was searchable before indexing started and the Magento store remains available for other requests. Final commit has much better performance than partial commit, and does not require any additional Solr configuration as does engine autocommit. Indexing begins after all unneeded data is removed and new data is added. At that point, users see results from newly indexed data immediately.● Partial commit: All content is removed from Solr after you reindex the content search index and users at that time see no search results. As content is gradually reindexed, users see only the results of content that has been indexed.● Engine autocommit: Content is put in the index queue but is not committed. You must configure Solr to commit at regular intervals (for example, every 5 minutes) or when a certain number of uncommitted items is reached. For more information, see the discussion of the <code><autoCommit></code> XML element in the solrconfig.xml section of the Solr Wiki. |

The following figure shows an example.

Catalog Search

| | |
|---|---|
| Minimal Query Length | <input type="text" value="1"/> [STORE VIEW] |
| Maximum Query Length | <input type="text" value="128"/> [STORE VIEW] |
| Maximum Query Words Count | <input type="text" value="10"/> [STORE VIEW] ▲ Applies for "Like" search type only. |
| Search Engine | <input type="text" value="Solr"/> [GLOBAL] |
| Solr Server Hostname | <input type="text" value="www.example.com"/> [GLOBAL] |
| Solr Server Port | <input type="text" value="8983"/> [GLOBAL] |
| Solr Server Username | <input type="text" value="admin"/> [GLOBAL] |
| Solr Server Password | <input type="text"/> [GLOBAL] |
| Solr Server Timeout | <input type="text" value="15"/> [GLOBAL] |
| Solr Server Path | <input type="text" value="solr"/> [GLOBAL] ▲ Without starting and trailing slashes. |
| Indexation Mode | <input type="text" value="Final commit"/> [GLOBAL] ▲ Note: Partial commit mode affects time of indexation, but allows indexed products to be visible on the frontend before indexation is completed. |
| Test Connection | |
| Enable Search Suggestions | <input type="text" value="Yes"/> [STORE VIEW] ▲ Note: Enabling this option affects the performance. |
| Search Suggestions Count | <input type="text" value="2"/> [STORE VIEW] |
| Show Results Count for Each Suggestion | <input type="text" value="No"/> [STORE VIEW] ▲ Note: Enabling this option affects the performance. |
| Enable Search Recommendations | <input type="text"/> [STORE VIEW] ▲ Note: Enabling this option affects the performance. |
| Enable Solr Engine for Catalog Navigation | <input type="text" value="Yes"/> [GLOBAL] |

e. Click **Test Connection**.

The button changes as follows.

| Button state | Meaning |
|---------------------------------------|--|
| Successful Test again? | The test connection succeeded. Click Save Config and continue with the next section. |
| Connection failed! Test again? | <p>The test connection failed. Try the following:</p> <ul style="list-style-type: none"> ● Examine the command window in which you started Solr for stack traces and exceptions. You must resolve those before you continue. In particular, make sure you started Solr as a user with <code>root</code> privileges. ● Verify that <code>allow_url_fopen = On</code> is present in your server's <code>php.ini</code>. If you are not sure where <code>php.ini</code> is located, you can create a <code>phpinfo.php</code> page to locate it. ● Verify the value of the Solr Server Hostname field. Make sure the server is available. You can try the server's IP address instead. ● Use the command <code>netstat -an grep listen-port</code> command to verify that the port specified in the Solr Server Port field is not being used by another process. For example, to see if Solr is running on its default port, use the following command: <pre>netstat -an grep 8983</pre> <p>If Solr is running on port 8983, it displays similar to the following:</p> <pre>tcp 0 0 ::::8983 ::::* LISTEN</pre> |

- If Solr is installed on a remote machine, use the `ping` command to verify that machine is reachable from your Magento instance.
- If SELinux is enabled, make sure the Solr servlet container's listen port is available; otherwise, Magento cannot communicate with the servlet container. For example, you can consult the [SELinux Centos wiki](#).

f. Only after the test connection succeeds, click **Save Config** and continue with the next section.

Basic Solr Configuration

This section discusses how to configure Magento to work with Solr using options in the Admin Panel. Although additional Solr customization is possible, it is beyond the scope of this guide.



Important: Customize the Solr search engine at your own risk. Magento supports only the options displayed in the Admin Panel. Customizing the Solr engine itself, while potentially useful, can cause issues with Magento. If you encounter problems with your customizations, do not contact Magento Support; instead, consult the resources available from the [Apache Solr Wiki](#).

To configure Magento to work with Solr:

- a. Start the Magento Admin Panel and log in as an administrator.
- b. Click **System > Configuration**.
- c. In the left navigation bar, under the CATALOG group, click **Catalog > Catalog Search**.
- d. The following table shows the minimum amount of information to enter to test the connection to your Solr search engine. Leave all other values at their defaults.

| Option | Description |
|---|--|
| Minimal Query Length | Enter the minimum number of characters permitted for a catalog search. |
| Maximum Query Length | Enter the maximum number of characters permitted for a catalog search. |
| Search Engine Solr Server Hostname Solr Server Port Solr Server Username Solr Server Password Solr Server Timeout Solr Server Path Indexation Mode | Discussed in Configuring Solr to Work With Magento . |
| Enable Search Suggestions | <p>Suggestions are the native Solr mechanism of advising users in the event they enter incomplete or incorrect user input. Suggestions, when enabled, are automatically provided as part of any search request.</p> <p>Solr completes incomplete or incorrect input using a dictionary that is based on the main index (and can be customized using configuration files to use any other arbitrary dictionary). Suggestions display with default text "Did you mean:" in the search results page if needed.</p> <p>Notes:</p> <ul style="list-style-type: none"> • Search suggestions are not the same as AJAX hints. • Enabling suggestions negatively affects performance because they result in more complex queries to Solr. |
| Search Suggestions Count | Enter the maximum number of suggestions to return. |
| Show Results Count for Each Suggestion | The default option, No , displays only the suggestion and not the number of results |

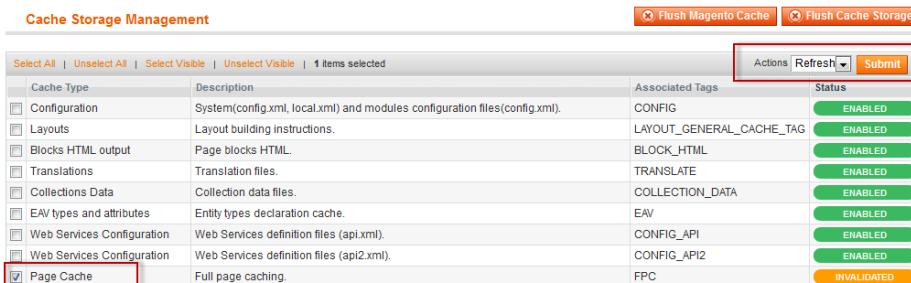
| | |
|--|---|
| | <p>for each suggestion.</p> <p>Click Yes to display the number of results for each suggestion.</p> |
| Enable Search Recommendations | <p><i>Recommendations</i> display terms related to a requested word or phrase on the search results page.</p> <p>This functionality is not based on third party engine functionality, but is implemented as part of the <code>Enterprise_Search</code> module and can be shown with the Solr search suggestions block.</p> <p>By default, Magento uses the <code>Enterprise_Search_Model_Adapter_HttpStream</code> module for recommendations. If you install the Apache Solr PHP extension, Magento automatically uses the <code>Enterprise_Search_Model_Adapter_PhpExtension</code> adapter instead. Both adapters function in the same way with no difference in performance. However, the <code>PhpExtension</code> adapter is not tested by Magento so you must thoroughly test any modifications you make to it before deploying it in a production environment.</p> <div style="background-color: #f4a460; padding: 10px; border-radius: 5px;">  Note: Enabling recommendations negatively affects performance because they result in more complex queries to Solr and more database calls. </div> |
| Search Recommendations Count | Enter the maximum number of recommendations to return. |
| Show Results Count for Each Recommendation | <p>The default option, No, displays only the recommendation and not the number of results for each recommendation.</p> <p>Click Yes to display the number of results for each recommendation.</p> |
| Enable Solr Engine for Catalog Navigation | <p>Click Yes (the default) to use Solr to enable layered navigation in the category view.</p> <p>Click No to use the database for layered navigation in the category view.</p> |

Reindexing Catalog Search and Refreshing the Full Page Cache

After you change the Solr configuration, you must reindex the catalog search index and refresh the full page cache as follows:

- In the Admin Panel, click **System > Cache Management**.
- Select the checkbox next to **Page Cache**.
- From the **Actions** list in the upper right, click **Refresh**.

The following figure shows an example.



The screenshot shows a grid titled "Cache Storage Management". At the top, there are buttons for "Flush Magento Cache" and "Flush Cache Storage". Below the buttons, there are links for "Select All", "Unselect All", "Select Visible", "Unselect Visible", and "1 items selected". The grid has columns for "Cache Type", "Description", "Associated Tags", and "Status". A red box highlights the "Page Cache" row under "Cache Type". The "Status" column for the highlighted row shows an orange "INVALIDATED" status, while other rows show "ENABLED".

| Cache Type | Description | Associated Tags | Status |
|--|--|--------------------------|-------------|
| Configuration | System(config.xml, local.xml) and modules configuration files(config.xml). | CONFIG | ENABLED |
| Layouts | Layout building instructions. | LAYOUT_GENERAL_CACHE_TAG | ENABLED |
| Blocks HTML output | Page blocks HTML. | BLOCK_HTML | ENABLED |
| Translations | Translation files. | TRANSLATE | ENABLED |
| Collections Data | Collection data files. | COLLECTION_DATA | ENABLED |
| EAV types and attributes | Entity types declaration cache. | EAV | ENABLED |
| Web Services Configuration | Web Services definition files (api.xml). | CONFIG_API | ENABLED |
| Web Services Configuration | Web Services definition files (api2.xml). | CONFIG_API2 | ENABLED |
| <input checked="" type="checkbox"/> Page Cache | Full page caching. | FPC | INVALIDATED |

- To update the catalog search index, open a command prompt window.
- Change to the `shell` subdirectory of your Magento installation directory.

For example, on CentOS:

```
cd /var/www/html/magento/shell
```

f. Enter the following command:

```
php indexer.php --reindex catalogsearch_fulltext
```

Scripting Solr Startup and Shutdown

In a production environment, you should start and stop Solr using a script.



Note: You must perform all tasks discussed in this section as a user with `root` privileges.

Create a script named `/etc/init.d/solr` with options similar to the following:

```
#!/bin/sh

#Starts, stops, and restarts Apache Solr.
#chkconfig: 35 92 08
#description: Starts and stops Apache Solr

$SOLR_DIR="[your Solr install dir]"
JAVA_OPTIONS="-Xmx1024m -DSTOP.PORT=jetty-stop-port -DSTOP.KEY=jetty-stop-key -jar start.jar"
LOG_FILE="path-to-solr-log-file"
JAVA="java_home"

case $1 in
start)
echo -n "Starting Solr"
cd $SOLR_DIR
$JAVA $JAVA_OPTIONS 2> $LOG_FILE &
;;
stop)
echo -n "Stopping Solr"
cd $SOLR_DIR
$JAVA $JAVA_OPTIONS --stop
;;
restart)
$0 stop
sleep 1
$0 start
;;
*)
echo "Usage: $0 {start|stop|restart}" >&2
exit 1
;;
esac
```

All parameters shown in the following table are required.

| Value | Description |
|---|---|
| [your Solr install dir] | The absolute file system path to your Solr installation. (For example, <code>/etc/solr/apache-solr-3.6.2</code>) |
| <code>jetty-stop-port</code> <code>jetty-stop-key</code> | Security parameters used to prevent malicious attempts to stop Jetty. For <code>-DSTOP.PORT=</code> , specify any unused port. For <code>-DSTOP.KEY=</code> , specify a string. If you omit a value for <code>-DSTOP.KEY=</code> , Jetty generates a random key you must enter to stop Jetty. For more information, see Securing Jetty . |

| | |
|------------------------------|--|
| <i>path-to-solr-log-file</i> | Absolute file system path to the Solr log file. (For example, /var/log/solr.log) |
| <i>java_home</i> | Absolute file system path to your Java executable. (For example, /usr/bin/java) |

An example follows:

```
#!/bin/sh

#Starts, stops, and restarts Apache Solr.
#chkconfig: 35 92 08
#description: Starts and stops Apache Solr

$SOLR_DIR="/etc/solr/apache-solr-3.6.2/example"
JAVA_OPTIONS="-Xmx1024m -DSTOP.PORT=8079 -DSTOP.KEY=mykey -jar start.jar"
LOG_FILE="/var/log/solr.log"
JAVA="/usr/bin/java"

case $1 in
start)
echo -n "Starting Solr"
cd $SOLR_DIR
$JAVA $JAVA_OPTIONS 2> $LOG_FILE &
;;
stop)
echo -n "Stopping Solr"
cd $SOLR_DIR
$JAVA $JAVA_OPTIONS --stop
;;
restart)
$0 stop
sleep 1
$0 start
;;
*)
echo "Usage: $0 {start|stop|restart}" >&2
exit 1
;;
esac
```

To complete the script:

- Make sure you saved the edited version of the script.
- Give the script executable permissions as follows:

```
chmod +x /etc/init.d/solr
```

- You can now start, stop, and restart Solr as follows:
 - Start Solr: /etc/init.d/solr start
 - Stop Solr: /etc/init.d/solr stop
 - Restart Solr: /etc/init.d/solr restart

Next Steps

For additional information about Solr, see the following:

- For more information about performance, see "Using Solr as a Search Engine" in [Maximizing Performance and Scalability with Magento Enterprise Edition](#)
- [Customize Solr](#)

Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

Information About Enterprise Edition (EE) Patches for Apache Solr

This article lists patches that Magento Support has made available for Magento EE versions 1.10.1.0 and later. The patches themselves are available on the Magento EE support portal.

To get the patches:

1. Log in to <http://magento.com> with your Magento EE credentials.
2. Click **Downloads** in the left navigation.
3. In the right pane, expand **Magento Enterprise Edition 1.x**.
4. Expand **Support Patches and Security Patches > Solr Search Engine patches**.

Following is a list of available patches:

- [Magento EE Solr Patch for EE 1.12.0.x \(ID: SUPEE-111\)](#)
- [Magento EE Solr Patch for EE 1.11.1.0 \(ID: SUPEE-107\)](#)
- [Magento EE Solr Patch for EE 1.11.0.x \(ID: SUPEE-108\)](#)
- [Magento EE Solr Patch for EE 1.10.1.x \(ID: SUPEE-110\)](#)

Magento EE Solr Patch for EE 1.12.0.x (ID: SUPEE-111)

Magento EE Version(s) Affected

This fix affects Magento EE 1.12.0.0, 1.12.0.1, and 1.12.0.1

Issues Fixed in the Magento EE 1.12.0.x Patch

- Using layered navigation filtering returns the wrong results. (For example, filtering by brand has no effect.)
- Resolved an issue where search results don't display correctly after a Magento upgrade.
- Catalog navigation works properly.
- Products display as expected in categories if the products have a Date attribute with the option **Used for Sorting in Product Listing** set to **Yes**. There are no exceptions in Magento logs after reindexing.
- Corrected the sort order of products searched by SKU.
- Search results of products with names and/or SKUs that contains numbers, letters, and a hyphen character (-) are as expected.
- Resolved issues with search results for products in a locale other than en_US with numeric SKUs.
- Resolved issues with Solr not returning product search results.
- Search results no longer include products that are either Disabled or Out of Stock.

Installation Instructions

To install this fix:

1. Stop Solr if it's currently running.
2. Download the patch .sh and transfer it to your Magento installation root directory.
For example, /var/www/html/magento.
3. Enter the following command as a user with sufficient privileges to write to Magento files (typically, the web server user or root):

```
sh PATCH_SUPEE-111_EE_1.12.0.0_v6.sh
```

Note: The patch version—indicated by _v6 in this example—is subject to change. The version you get might be different.

A message such as the following displays to confirm the patch installed successfully:

Patch was applied/reverted successfully.

4. To reapply ownership to the files changed by the patch:
 - a. Find the web server user: ps -o "user group command" -C httpd,apache2
The value in the USER column is the web server username.
Typically, the Apache web server user on CentOS is apache and the Apache web server user on Ubuntu is www-data.
 - b. As the owner of the files in the Magento installation directory or as a user with root privileges, enter the following command from the Magento installation directory:
chown -R web-server-user-name . For example, on Ubuntu where Apache usually runs as www-data, enter chown -R www-data .
5. Copy the updated configuration files from Magento to your Solr configuration directory.
 - a. Back up the files in your current Solr configuration directory.
 - b. Enter the following command:

```
cp magento-install-root/lib/Apache/Solr/conf/* solr-config-dir
```

For example, if Magento is installed in /var/www/magento and Solr configuration files are located in /etc/apache-solr-3.5.0/mysqlrapp/solr/conf, enter

```
cp /var/www/magento/lib/Apache/Solr/conf/* /etc/apache-solr-3.5.0/mysqlrapp/solr/conf
```
6. Start Solr.
7. Log in to the Admin Panel as an administrator.
8. Click **System > Cache Management**.
9. Click **Flush Magento Cache**.
10. Click **System > Configuration**.
11. In the left navigation bar, from the CATALOG group, click **Catalog**.
12. In the right pane, expand **Catalog Search**.
13. From the **Indexation Mode** list, click **Final Commit**.
This is the setting Magento recommends. For a description of the other options, see the [Magento EE User Guide](#).
14. Click **System > Index Management**.
15. Click the **Reindex Data** link in the Catalog Search row.
16. Reindex other indexers if necessary.

Reverting the Patch

If applying the patch results in errors, [contact Magento Support](#). If you are instructed to do so, revert the patch:

1. Change to your Magento installation directory.
2. Enter the following command as a user with sufficient privileges to write to Magento files (typically, the web server user or root):

```
sh PATCH_SUPEE-111_EE_1.12.0.0_v6.sh -R
```

Troubleshooting

If you get an error when you run the patch, use the following suggestions:

- Verify the patch is located in your Magento installation root directory.
Ubuntu example: /var/www/magento
CentOS example: /var/www/html/magento
- Verify you're running the patch with sufficient privileges.
Typically, this means running it as the web server user or as a user with root privileges.

- Try running the patch again.
If problems persist, contact [Magento Support](#).

Magento EE Solr Patch for EE 1.11.1.0 (ID: SUPEE-107)

Magento EE Version(s) Affected

This fix affects Magento EE 1.11.1.0.

Issues Fixed in the Magento EE 1.11.1.0 Patch

- Search results display correctly if products have attribute values set at the store view.
- You can now enable or disable search engine recommendations.
- Search results properly detect the configured price navigation step calculation.
- Reindexing performance has been improved.
- Solr search displays expected results when products use an attribute set for **Catalog Input Type for Store Owner** configured for **Price**—and this attribute is used for sorting. The following error no longer displays after reindexing the catalog search index:
`SEVERE: org.apache.solr.common.SolrException: can not sort on multivalued field`
- Search results display in the order consistent with the setting for **Catalog > Attributes > Manage Label > Options**.
- Search results display correctly if you use non-default attributes.
- A product that uses an attribute name that has a trailing space displays in search results.
- Resolved an issue that prevented products from displaying in the web store if Solr is enabled.
- Products can be located using layered navigation if the products have attributes set for **Use in Quick Search** and/or **Use in Advanced Search** set to **No**.
- Advanced search works properly when Solr is enabled.
- Category permissions work properly when Solr is enabled.
- Changing pricing, such as configuring a discount for an existing product, automatically re-indexes and displays the correct price in the web store.
- Advanced search results display as expected after toggling the value of a Yes/No attribute.
- A product displays in search results after changing the value of the product's Visibility setting to **Catalog**.
- Quick search results display as expected when the value of maximum query length is null.
- Searching for the value of a Yes/No attribute in a supported locale other than en_US works properly.
- Resolved issues with incorrect attribute values being assigned to a configurable product.
- Out of stock products display in the web store when **Display Out of Stock Products** is set to **Yes**.
- Products display in advanced search if they have attributes of multiple select and drop-down.
- Quick search works properly in store views with a locale other than English.
- Cron job reindexing uses the specified indexation mode for the catalog search index.
- Simple products that are configured as part of a grouped product display in search results for the simple product's SKU.
- A product's price attribute displays in layered navigation when Solr is enabled.
- Resolves an issue with attributes displaying in layered navigation as *empty*.
- Catalog navigation works properly.
- Resolved issues with Solr after upgrading to EE 1.11.1.0
- Products display in categories after import.

Installation Instructions

To install this fix:

1. Stop Solr if it's currently running.
2. Download the patch .sh and transfer it to your Magento installation root directory.
For example, /var/www/html/magento.

3. Enter the following command as a user with sufficient privileges to write to Magento files (typically, the web server user or `root`):

```
sh PATCH_SUPEE-107_EE_1.11.1.0_v1.sh
```

Note: The patch version—indicated by `_v6` in this example—is subject to change. The version you get might be different.

A message such as the following displays to confirm the patch installed successfully:

`Patch was applied/reverted successfully.`

4. To reapply ownership to the files changed by the patch:

- Find the web server user: `ps -o "user group command" -C httpd,apache2`
The value in the USER column is the web server username.

Typically, the Apache web server user on CentOS is `apache` and the Apache web server user on Ubuntu is `www-data`.

- As the owner of the files in the Magento installation directory or as a user with `root` privileges, enter the following command from the Magento installation directory:

`chown -R web-server-user-name .` For example, on Ubuntu where Apache usually runs as `www-data`, enter `chown -R www-data .`

5. Copy the updated configuration files from Magento to your Solr configuration directory.

- Back up the files in your current Solr configuration directory.

- Enter the following command:

```
cp magento-install-root/lib/Apache/Solr/conf/* solr-config-dir
```

For example, if Magento is installed in `/var/www/magento` and Solr configuration files are located in `/etc/apache-solr-3.5.0/mysqlapp/solr/conf`, enter

```
cp /var/www/magento/lib/Apache/Solr/conf/* /etc/apache-solr-3.5.0/mysqlapp/solr/conf
```

6. Start Solr.

7. Log in to the Admin Panel as an administrator.

8. Click **System > Cache Management**.

9. Click **Flush Magento Cache**.

10. Click **System > Configuration**.

11. In the left navigation bar, from the CATALOG group, click **Catalog**.

12. In the right pane, expand **Catalog Search**.

13. From the **Indexation Mode** list, click **Final Commit**.

This is the setting Magento recommends. For a description of the other options, see the [Magento EE User Guide](#).

14. Click **System > Index Management**.

15. Click the **Reindex Data** link in the Catalog Search row.

16. Reindex other indexers if necessary.

Reverting the Patch

If applying the patch results in errors, [contact Magento Support](#). If you are instructed to do so, revert the patch:

1. Change to your Magento installation directory.

2. Enter the following command as a user with sufficient privileges to write to Magento files (typically, the web server user or `root`):

```
sh PATCH_SUPEE-107_EE_1.11.1.0_v1.sh -R
```

Troubleshooting

If you get an error when you run the patch, use the following suggestions:

- Verify the patch is located in your Magento installation root directory.

Ubuntu example: `/var/www/magento`

CentOS example: `/var/www/html/magento`

- Verify you're running the patch with sufficient privileges.

Typically, this means running it as the web server user or as a user with `root` privileges.

- Try running the patch again.

If problems persist, [contact Magento Support](#).

Magento EE Solr Patch for EE 1.11.0.x (ID: SUPEE-108)

Magento EE Version(s) Affected

This fix affects Magento EE 1.11.0.0, 1.11.0.1, and 1.11.0.2.

Issues Fixed in the Magento EE 1.11.0.x Patch

- Solr does not display an anchor category if the layered navigation block is absent, such as when this block has been removed from the layout. Search results in this case are processed by the database, not by Solr.
- Search results display correctly regardless of the order in which content blocks load.
- Paginated search results display the total number of pages returned by a Solr search.
- Search results display properly for all locales supported by the Magento configuration. (A *supported locale* is one for which you have uploaded translations.)
- Reindexing using a cron job no longer causes PHP fatal errors.
- You can now enable or disable search engine recommendations.
- Search results properly detect the configured price navigation step calculation.
- Solr search displays expected results when products use an attribute set for **Catalog Input Type for Store Owner** configured for **Price**—and this attribute is used for sorting. The following error no longer displays after reindexing the catalog search index:
`SEVERE: org.apache.solr.common.SolrException: can not sort on multivalued field`
- Reindexing performance has been improved.
- Search results display in the order consistent with the setting for **Catalog > Attributes > Manage Label > Options**.
- Search results display correctly if you use non-default attributes.
- Products that use an attribute name that has a trailing space display in search results.
- Resolved an issue that prevented products from displaying in the web store if Solr is enabled.
- Advanced search works properly when Solr is enabled.
- Category permissions work properly when Solr is enabled.
- Changing pricing, such as configuring a discount for an existing product, automatically re-indexes and displays the correct price in the web store.
- Advanced search results display as expected after toggling the value of a Yes/No attribute.
- A product displays in search results after changing the value of the product's Visibility setting to **Catalog**.
- Searching for the value of a Yes/No attribute in a supported locale other than en_US works properly.
- Resolved issues with incorrect attribute values being assigned to a configurable product.
- Out of stock products display in the web store when **Display Out of Stock Products** is set to **Yes**.
(In the Admin Panel, click **System > Configuration > CATALOG > Inventory > Stock Options**.)
- Products display in advanced search if they have attributes of multiple select and drop-down.
- Quick search works properly in store views with a locale other than English.
- Products can be located using layered navigation if the products have attributes set for **Use in Quick Search** and/or **Use in Advanced Search** set to **No**.
- Quick search results display as expected when the value of maximum query length is null.
(In the Admin Panel, click **System > Configuration CATALOG > Catalog Search**, value of **Maximum Query Length**.)
- Cron job reindexing uses the specified indexation mode for the catalog search index.
- Simple products that are configured as part of a grouped product display in search results for the simple product's SKU.
- A product's price attribute displays in layered navigation when Solr is enabled.
(Attribute is configured with **Catalog Input Type for Store Owner** set to **Price** and **Use In Layered Navigation** set to **Filterable (with results)**.)
- Products display as expected in categories if the products have a Date attribute with the option **Used for Sorting in Product Listing** set to **Yes**. There are no exceptions in Magento logs after reindexing.
- Resolves an issue with attributes displaying in layered navigation as *empty*.
- Resolved an issue where search results don't display correctly after a Magento upgrade.
- Catalog navigation works properly.
- Search results display correctly if products have attribute values set at the store view.

Installation Instructions

To install this fix:

1. Stop Solr if it's currently running.
2. Download the patch .sh and transfer it to your Magento installation root directory.
For example, /var/www/html/magento.
3. Enter the following command as a user with sufficient privileges to write to Magento files (typically, the web server user or root):

```
sh PATCH_SUPEE-108_EE_1.11.0.2_v1.sh
```

Note: The patch version—indicated by _v1 in this example—is subject to change. The version you get might be different.

A message such as the following displays to confirm the patch installed successfully:

```
Patch was applied/reverted successfully.
```

4. To reapply ownership to the files changed by the patch:
 - a. Find the web server user: ps -o "user group command" -C httpd,apache2
The value in the USER column is the web server username.
Typically, the Apache web server user on CentOS is apache and the Apache web server user on Ubuntu is www-data.
 - b. As the owner of the files in the Magento installation directory or as a user with root privileges, enter the following command from the Magento installation directory:
chown -R web-server-user-name .
For example, on Ubuntu where Apache usually runs as www-data, enter
chown -R www-data .
5. Copy the updated configuration files from Magento to your Solr configuration directory.
 - a. Back up the files in your current Solr configuration directory.
 - b. Enter the following command:
cp magento-install-root/lib/Apache/Solr/conf/* solr-config-dir
For example, if Magento is installed in /var/www/magento and Solr configuration files are located in /etc/apache-solr-3.5.0/mysqlrapp/solr/conf, enter
cp /var/www/magento/lib/Apache/Solr/conf/* /etc/apache-solr-3.5.0/mysqlrapp/solr/conf
6. Start Solr.
7. Log in to the Admin Panel as an administrator.
8. Click **System > Cache Management**.
9. Click **Flush Magento Cache**.
10. Click **System > Configuration**.
11. In the left navigation bar, from the CATALOG group, click **Catalog**.
12. In the right pane, expand **Catalog Search**.
13. From the **Indexation Mode** list, click **Final Commit**.
This is the setting Magento recommends. For a description of the other options, see the [Magento EE User Guide](#).
14. Click **System > Index Management**.
15. Click the **Reindex Data** link in the Catalog Search row.
16. Reindex other indexers if necessary.

Reverting the Patch

If applying the patch results in errors, [contact Magento Support](#). If you are instructed to do so, revert the patch:

1. Change to your Magento installation directory.
2. Enter the following command as a user with sufficient privileges to write to Magento files (typically, the web server user or root):

```
sh PATCH_SUPEE-108_EE_1.11.0.2_v1.sh -R
```

Troubleshooting

If you get an error when you run the patch, use the following suggestions:

- Verify the patch is located in your Magento installation root directory.
Ubuntu example: /var/www/magento
CentOS example: /var/www/html/magento
- Verify you're running the patch with sufficient privileges.

Typically, this means running it as the web server user or as a user with `root` privileges.

- Try running the patch again.
If problems persist, contact [Magento Support](#).

Magento EE Solr Patch for EE 1.10.1.x (ID: SUPEE-110)

Magento EE Version(s) Affected

This fix affects Magento EE 1.10.1.0 and 1.10.1.1.

Issues Fixed in the Magento EE 1.10.1.x Patch

- Solr does not display an anchor category if the layered navigation block is absent, such as when this block has been removed from the layout. Search results in this case are processed by the database, not by Solr.
- Solr search results display correctly with different settings for an attribute set that uses **Use In Layered Navigation** set to **Filterable (no results)**.
In other words, if some products are assigned to a category that do *not* use that attribute set, they display correctly in Solr search results if there are products in other categories that *do* use the attribute set.
- Catalog search reindexing has better performance.
- Paginated search results display the total number of pages returned by a Solr search.
- Search results display properly for all locales supported by the Magento configuration. (A *supported locale* is one for which you have uploaded translations.)
- Search results properly detect the configured price navigation step calculation.
(In the Admin Panel, click **System > Configuration > CATALOG > Catalog > Layered Navigation**. From the **Price Navigation Step Calculation** list, click **Manual**.)
- Search results display correctly if you use non-default attributes.
- Relevancy and suggestions were improved.
- Catalog navigation works properly.
- When the user chooses to display all search results, the correct number of results display.
- Search results include only products that are in stock.

Installation Instructions

To install this fix:

1. Download the patch .sh and transfer it to your Magento installation root directory.
For example, `/var/www/html/magento`.
2. Enter the following command as a user with sufficient privileges to write to Magento files (typically, the web server user or `root`):

```
sh PATCH_SUPEE-110_EE_1.10.1.1_v2.sh
```

Note: The patch version—indicated by `_v2` in this example—is subject to change. The version you get might be different.

A message such as the following displays to confirm the patch installed successfully:

```
Patch was applied/reverted successfully.
```

3. To reapply ownership to the files changed by the patch:
 - a. Find the web server user: `ps -o "user group command" -C httpd,apache2`
The value in the USER column is the web server username.
Typically, the Apache web server user on CentOS is `apache` and the Apache web server user on Ubuntu is `www-data`.
 - b. As a user with `root` privileges, enter the following command from the Magento installation directory:
`chown -R web-server-user-name .`
For example, on Ubuntu where Apache usually runs as `www-data`, enter
`chown -R www-data .`
4. If Solr is running, stop it.
5. Copy the updated configuration files from Magento to your Solr configuration directory.
 - a. Back up the files in your current Solr configuration directory.
 - b. Enter the following command:
`cp magento-install-root/lib/Apache/Solr/conf/* solr-config-dir`
For example, if Magento is installed in `/var/www/magento` and Solr configuration files are located in `/etc/apache-solr-3.5.0/mysqlrapp/solr/conf`, enter

- ```
cp /var/www/magento/lib/Apache/Solr/conf/* /etc/apache-solr-3.5.0/mysqlapp/solr/conf
```
6. Start Solr.
  7. Log in to the Admin Panel as an administrator.
  8. Click **System > Cache Management**.
  9. Click **Flush Magento Cache**.
  10. Click **System > Configuration**.
  11. In the left navigation bar, from the CATALOG group, click **Catalog**.
  12. In the right pane, expand **Catalog Search**.
  13. From the **Indexation Mode** list, click **Final Commit**.  
This is the setting Magento recommends. For a description of the other options, see the [Magento EE User Guide](#).
  14. Click **System > Index Management**.
  15. Click the **Reindex Data** link in the Catalog Search row.
  16. Reindex other indexers if necessary.

## Reverting the Patch

If applying the patch results in errors, [contact Magento Support](#). If you are instructed to do so, revert the patch:

1. Change to your Magento installation directory.
2. Enter the following command as a user with sufficient privileges to write to Magento files (typically, the web server user or `root`):

```
sh PATCH_SUPEE-110_EE_1.10.1.1_v2.sh -R
```

## Troubleshooting

If you get an error when you run the patch, use the following suggestions:

- Verify the patch is located in your Magento installation root directory.  
Ubuntu example: `/var/www/magento`  
CentOS example: `/var/www/html/magento`
- Verify you're running the patch with sufficient privileges.  
Typically, this means running it as the web server user or as a user with `root` privileges.
- Try running the patch again.  
If problems persist, [contact Magento Support](#).

 2 months ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento Enterprise Edition (EE) 1.13 Benchmarking Guide

## Contents

- [Customer Needs and Expectations](#)
- [Multi-Node Deployment Topology](#)
- [Scenarios](#)
- [Benchmarking Results](#)
- [Conclusion](#)
- [Appendix](#)

## Customer Needs and Expectations

The focus of the Magento Enterprise Edition 1.13 release is performance and scalability. The benchmarking results presented here demonstrate that we have addressed the following concerns:

1. The eCommerce landscape is changing, and merchants must provide customers with an online shopping experience that meets their performance expectations
2. As merchants grow their businesses and increasingly larger enterprise merchants are adopting Magento, Magento Enterprise Edition must scale to handle increased traffic volume and larger catalogs

## Magento's Plan

We identified the following areas for enhancement in Magento Enterprise Edition 1.13:

- Improve indexing for catalogs of all sizes
- Make re-indexing operations invisible to the shopper by executing them in the background
- Improve full page caching support
- Reduce page load times for key shopping flows (checkout)
- Enable merchants to serve heavier traffic volume without need to purchase additional hardware

## What Magento Accomplished

Magento Enterprise Edition 1.13 delivers the following improvements over Magento Enterprise Edition 1.12:

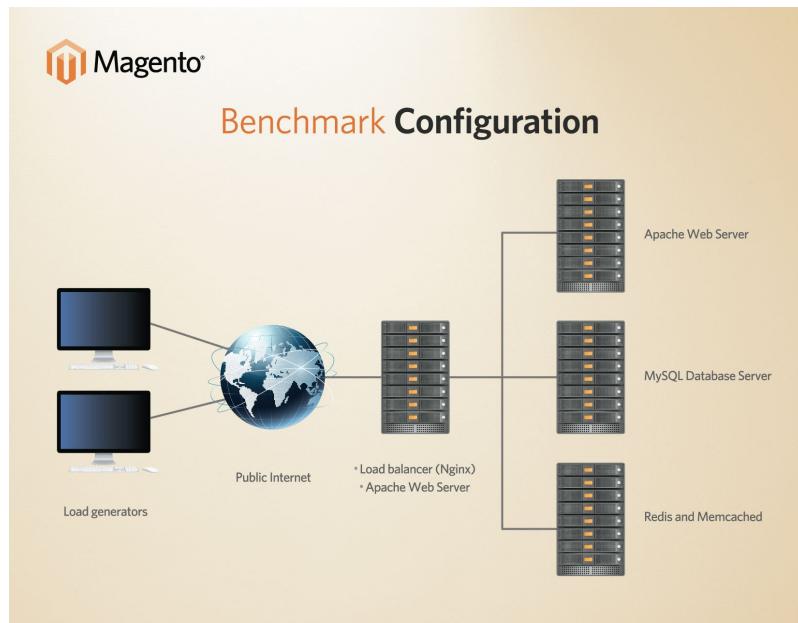
- Incremental re-indexing has been introduced with Magento Enterprise Edition 1.13, and scenarios where full-re-indexing was required have been limited. This means that operations that previously took hours can now be completed in minutes
- 53% improvement in completion times for a full re-index with a 500,000 SKU catalog
- 35% improvement in "Place Order" performance

- 65% improvement in page load times across shopper flow pages that were tested.
- 33% improvement in orders per day that can be processed on the same hardware configuration as Magento Enterprise Edition 1.12
- 31% improvement in page views per day supported on the same hardware configuration on Magento Enterprise Edition 1.12

## Multi-Node Deployment Topology

Before we introduce you to the benchmarks, an overview of our multi-node benchmarking facility is in order. We started with a basic multi-node cluster with load balancer and caching and separate DB node, and installed Apache. This is a familiar hardware configuration for the Magento developer community. Once provisioned, we installed both Magento Enterprise Edition 1.13 and Magento Enterprise Edition 1.12 on the cluster to compare their performance.

This is a representation of the multi-node Magento Enterprise Edition 1.12/1.13 installation used for performance testing. It consists of four physical nodes, three of which are virtualized.



We used HP ProLiant SL230s Gen8 servers with 8x16GB DDR3 and 2x Intel Xeon E5-2660 CPUs for our benchmarking.

We used a physical disk of a usable 1.2 TB managed by a RAID10 controller. The load balancer we used is nginx.

The cluster resides in our Las Vegas data center, connected to the Internet via a gigabit connection. We tested the cluster using the popular Gatling suite from our engineering offices in Austin, TX.

## Software Components

| Software Component           | Version | Notes                                                                                           |
|------------------------------|---------|-------------------------------------------------------------------------------------------------|
| Magento EE                   | 1.13    | Magento eCommerce server Enterprise Edition                                                     |
| Apache, mpm_prefork, mod_php | 2.2.15  | Industry-standard Web server plus modules for optimizing process control and PHP code execution |
| CentOS                       | 6.3     | Linux operating system                                                                          |
| MySQL                        | 5.5     | Relational store for Magento Application                                                        |
| PHP                          | 5.3.23  | Interpreter for Magento logic                                                                   |
| Memcached                    | 1.4.15  | Shared memory caching service                                                                   |
| Nginx                        | 1.2.7   | Load balancing service                                                                          |
| Redis                        | 2.6.11  | Data structure server (key-value pairs a.k.a. hashtables)                                       |

## Scenarios

This section presents the merchant scenarios we simulated for our testing. These scenarios are based on real-world experience and industry standards with respect to shopper flows and catalog sizes.

## Shopper Flows

We used real-world, established eCommerce metrics to simulate shopper flows.

- 94% of eCommerce shoppers visited a storefront but did not purchase any products
- 65% of shoppers added items to their carts but abandoned them
- Of the 6% who did purchase products (otherwise known as the conversion rate), half checked out as guests without signing in, and half signed into the storefront.

| Scenario                                   | Benchmark test traffic | Target traffic |
|--------------------------------------------|------------------------|----------------|
| Concurrent virtual users sessions          | 1000                   | 2000           |
| Checkouts completed as guest shopper       | 3%                     | 3%             |
| Checkouts completed as logged in shopper   | 3%                     | 3%             |
| Browse store, add to cart and abandon cart | 65%                    | 65%            |
| Conversion Rate                            | 3%                     | 3%             |
| Test duration as peak load                 | 10 minutes             | 4 hours        |

## Catalog

The catalogs we simulated also reflect real-world, established eCommerce experience.

- We simulated small and medium-sized companies with a 50,000-item catalog with 27 product categories.
- We simulated large companies 500,000 items in the catalog with 2,000 categories.
- As many eCommerce sites offer different types of products, we specified physical (simple) products, virtual products, and downloadable products, with 60% of the products in each catalog being physical.
- We benchmarked each catalog on one website/storefront.

| Requirements                       | Standard Merchant                | Large Merchant                   |
|------------------------------------|----------------------------------|----------------------------------|
| Catalog Size (products/categories) | 50K SKUs, 27 Categories          | 500K, 2000 Categories            |
| Product types                      | Simple, virtual and downloadable | Simple, virtual and downloadable |
| Distribution of product types      | 60%, 20%, 20%                    | 60%, 20%, 20%                    |
| Attribute range per product        | Default                          | Default                          |
| Attribute sets                     | Default                          | Default                          |
| Items in checkout carts            | 3                                | 3                                |
| # of cart Items on promotion       | 0                                | 0                                |
| Number of websites / store fronts  | 1                                | 1                                |

## Target Merchant Profile

The simulated merchant profile we benchmarked against represents the profile of our enterprise customers. We established these metrics based on the day-to-day experience of large merchants operating a successful eCommerce business. Again, we used what we consider to be a typical hardware and software configuration.

- 50K visitors / day
- 1M Page views / day
- 18000 orders / day
- 3000 orders during peak 4 hours

- 1000 concurrent users
- Standard HW/SW configuration

## Benchmarking Results

These benchmarks were generated during extensive testing of our multi-node configuration running Magento Enterprise Edition 1.13 described above. Our configuration was modeled after what a commercial hosting partner would put into production, and the results reflect accurate gains over Magento Enterprise Edition 1.12.

The duration of the test sessions is 72 seconds, which significantly stresses the multi-node configuration.

### Incremental Re-indexing

In Magento Enterprise Edition 1.12, any change to a product would result in a full re-index. Magento Enterprise Edition 1.13 introduces a new feature--incremental re-indexing. With incremental re-indexing, only those items that were changed or added will be re-indexed, reducing the processing time to a fraction of what was required before.

Take the example of a merchant with a catalog containing 500,000 products. In Magento Enterprise Edition 1.12, any change to a product would result in a full re-index operation. In Magento Enterprise Edition 1.13, incremental re-indexing means the merchant will only re-index items that were changed. The test focused on measuring the improvements provided by the incremental re-indexing feature in Magento Enterprise Edition 1.13. The table below compares improvements to common admin actions, such as changing a product description, prices or inventory.

| Action                     | EE 1.12    | EE 1.13<br>20 Products | EE 1.13<br>200 Products |
|----------------------------|------------|------------------------|-------------------------|
| Change Product Description | 84 minutes | 4 seconds              | 6 seconds               |
| Change Product Price       | 9 minutes  | 0.86 seconds           | 1.2 seconds             |
| Change inventory           | 56 seconds | 0.11 seconds           | 0.21 seconds            |

### Full Re-indexing

As part of the benchmarking effort, we also measured the improvements in the full re-indexing feature on Magento Enterprise Edition 1.13, where indexing a 500,000-item catalog was 53% faster than Magento Enterprise Edition 1.12. Faster re-indexing means less load on the system and that changes to the catalog propagate faster to the storefront.

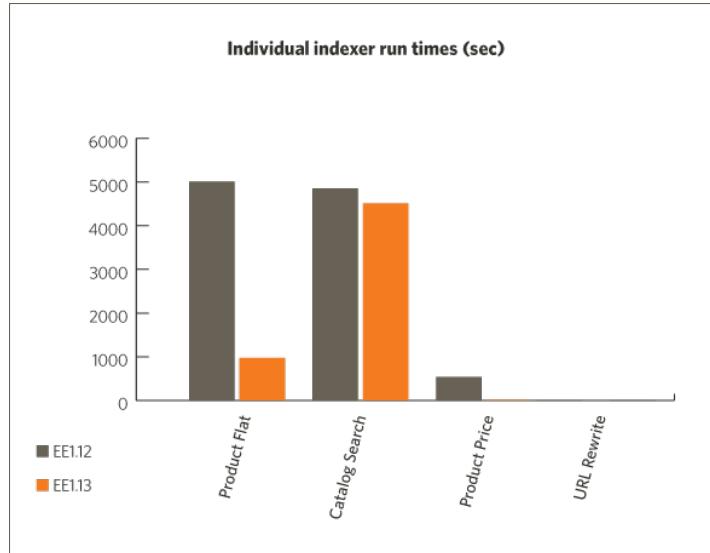
| Magento EE 1.12 | Magento EE 1.13 | % Improvement |
|-----------------|-----------------|---------------|
| 3:16:23         | 1:33:50         | 53%           |

- 500,000 products
- 2,000 categories
- One store
- One catalog update during test run

### Individual Re-indexers

The Magento Enterprise Edition 1.13 indexer component contains a number of individual indexers such as Product Flat Data and Product Price. When a Magento admin changes the price of a product, it is only necessary to execute the Product Price indexer for pricing changes to propagate to the frontend. The completion times of these individual indexers were measured in the benchmark environment for Magento Enterprise Edition 1.13 and Magento Enterprise Edition 1.12.

This section presents the results for full re-index completion times for the individual indexers in Magento Enterprise Edition 1.13 compared to Magento Enterprise Edition 1.12.



- URL Rewrite failed to run in Magento Enterprise Edition 1.12 but takes only 0.15 sec in Magento Enterprise Edition 1.13
- Catalog description: 500,000 products, 2000 categories, one storefront

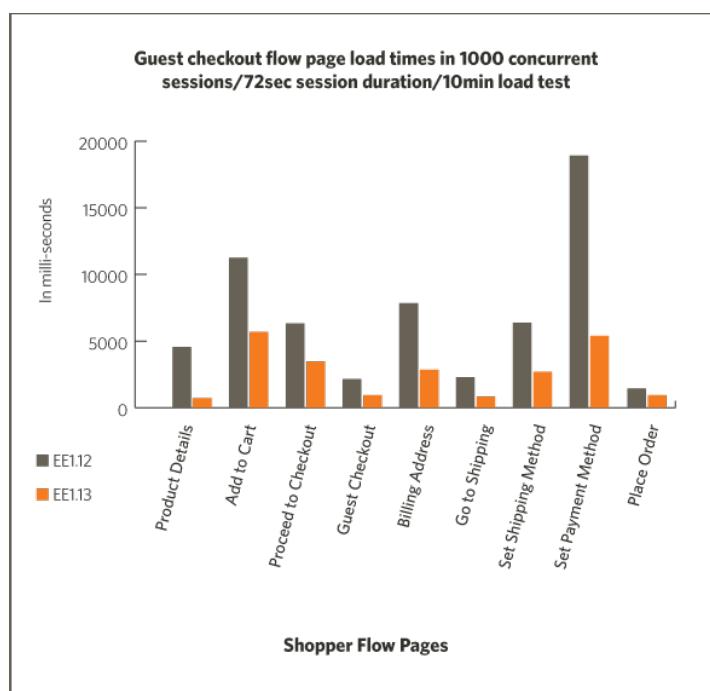
## Page Load Times

When Magento Enterprise Edition 1.12 and Magento Enterprise Edition 1.13 were compared, with both running on our multi-node benchmarking configuration, Magento Enterprise Edition 1.13 loaded pages 65% faster than Magento Enterprise Edition 1.12.

Guest checkout and registered user checkout are two flows that are crucial to storefront operation. This section presents the results of page load time measurements for these two flows.

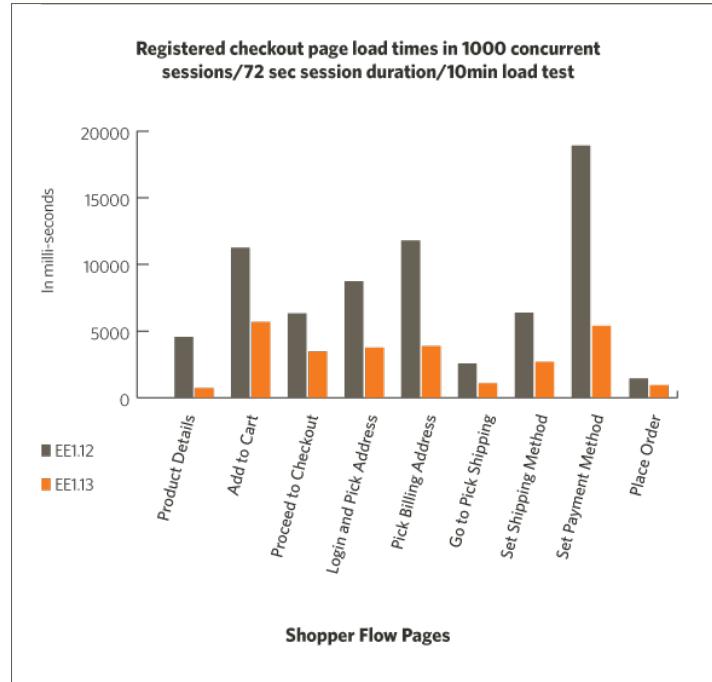
### Guest Checkout Flow Page Load Times

In the guest checkout flow pages, Magento Enterprise Edition 1.13 provides a substantial decrease in load times over its predecessor, most of the time more than twice as fast.



### Registered Checkout Flow Page Load Times

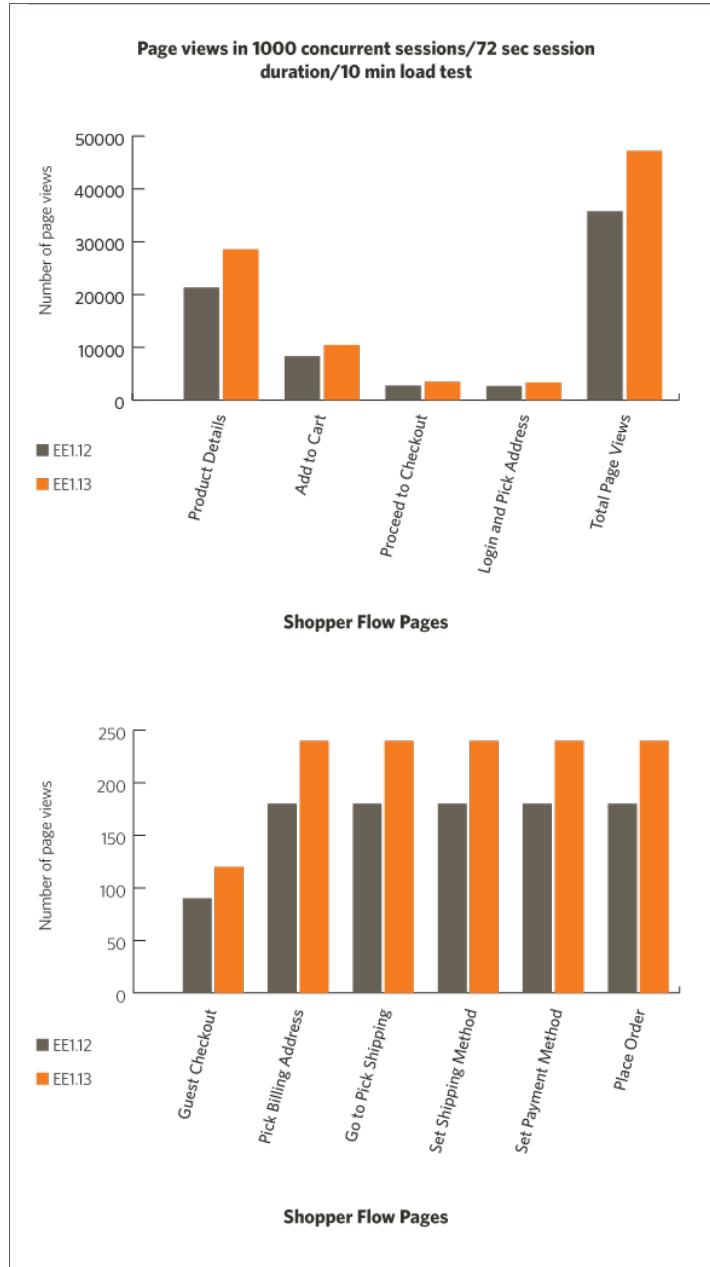
The bar chart below presents the improvements in page load times for registered checkout flow.



## Page Views and Orders

In addition to page load times, the benchmark also focused on measuring throughput improvements, particularly page views per day and orders per day.

During our testing, which simulated a storefront running at peak hours, EE 1.13 executed 33% more orders and 31% more page views than Magento Enterprise Edition 1.12 on the multi-node benchmarking configuration. Notably, Magento Enterprise Edition 1.13 served 47K pages during the test run (10 minutes).



- 10 minute session time

## Observations

What we noted during the benchmarking tests:

- The MySQL instance did not show any significant signs of CPU or I/O load during the tests.
- • The CPU was under 10% and no queries exceeded a 2-second threshold.
- Redis and Memcached instances did not exceed a CPU load of 10% during the tests.
- Web nodes showed high levels of CPU utilization under high load.
- We anticipate achieving stable scaling by adding additional web nodes to the cluster until the services themselves begin to degrade.

## Conclusion

Magento Enterprise Edition 1.13 was engineered for performance--and clearly delivers on the goal as measured by important metrics.

- Indexing is improved to enable faster operations without impacting shopping experience. Merchant administrators can add and update products as needed while ensuring product URLs, promotions, navigational menus, and product search tools are always up to date.
- The checkout process is improved by reducing page load times for browsing and placing orders. Faster checkout can significantly improve your customers' shopping experience and customer satisfaction, and potentially improve your conversion rate.
-

Faster page load times means Magento Enterprise Edition 1.13 can support more page views per day and more orders per day, potentially increasing your conversion rate.

In addition, we have focused on providing these benefits without the need to upgrade existing hardware, improving your return on assets and investment.

## Appendix

### Page Load Times

This benchmark report presented analysis of the results produced by the Gatling load generator tool. In this appendix, you can see the actual results reported by Gatling. The results were obtained under the following test setup, which was designed to push Magento to its limit.

|                            |            |
|----------------------------|------------|
| Concurrent session (users) | 1000       |
| Peak load duration         | 10 minutes |
| Session duration           | 72 seconds |
| CPU utilization            | 95%        |

CPU utilization is high because session duration was set to 72 seconds for the test. Session duration was set to a low value to increase the active load on Magento. A typical e-commerce site will see session durations of five minutes or more. Preliminary experiments by the benchmark team have confirmed that using a five-minute session duration reduces the CPU utilization to 50-60%.

| Magento EE v1.13                                       | Total | OK    | KO | Min  | Max   | Mean | Std Dev | 95th Pct | 99th Pct | Req/s |
|--------------------------------------------------------|-------|-------|----|------|-------|------|---------|----------|----------|-------|
| Requests                                               |       |       |    |      |       |      |         |          |          |       |
| Global Information0                                    | 62161 | 62087 | 74 | 80   | 12910 | 1735 | 1869    | 5780     | 8060     | 75    |
| product page0                                          | 28573 | 28573 | 0  | 80   | 9390  | 709  | 996     | 2670     | 4920     | 34    |
| Click add to cart1                                     | 10458 | 10458 | 0  | 440  | 7170  | 1868 | 1139    | 4010     | 4980     | 13    |
| Click add to cart Redirect 12                          | 10458 | 10458 | 0  | 960  | 12910 | 3816 | 2239    | 8050     | 9960     | 13    |
| click checkout3                                        | 3486  | 3486  | 0  | 1100 | 10800 | 3459 | 1965    | 7130     | 8970     | 4     |
| Registered Checkout:Login and Pick Address4            | 3366  | 3366  | 0  | 200  | 7380  | 1155 | 1482    | 4870     | 5900     | 4     |
| Registered Checkout:Login and Pick Address Redirect 15 | 3366  | 3366  | 0  | 700  | 11440 | 2605 | 1662    | 5640     | 7710     | 4     |
| Guest Checkout start6                                  | 120   | 120   | 0  | 360  | 2490  | 908  | 487     | 1860     | 2420     | 0     |
| Registered:Pick Billing Address 17                     | 120   | 120   | 0  | 390  | 4240  | 1457 | 974     | 3510     | 4210     | 0     |
| Registered:Pick Billing Address 2-18                   | 120   | 120   | 0  | 410  | 3690  | 1218 | 685     | 2460     | 3070     | 0     |
| Registered:Pick Billing Address 2-29                   | 120   | 120   | 0  | 400  | 3600  | 1201 | 714     | 2700     | 3510     | 0     |
| Registered:Go to pick Shipping Method10                | 120   | 120   | 0  | 340  | 2900  | 1072 | 633     | 2330     | 2700     | 0     |
| Guest:Pick Billing Address 111                         | 120   | 120   | 0  | 610  | 4620  | 1688 | 941     | 3580     | 4440     | 0     |
| Guest:Pick Billing Address 212                         | 120   | 120   | 0  | 380  | 3530  | 1049 | 589     | 2050     | 2750     | 0     |
| Guest:Go to pick Shipping Method13                     | 120   | 120   | 0  | 300  | 2080  | 796  | 401     | 1530     | 1960     | 0     |
| Set Shipping Method (Flatrate) 114                     | 240   | 240   | 0  | 510  | 4830  | 1780 | 1066    | 3930     | 4720     | 0     |
| Set Shipping Method (Flatrate): Goto Payment15         | 240   | 240   | 0  | 300  | 3150  | 905  | 516     | 1940     | 2300     | 0     |
| Set Payment Method (Check/MO) 116                      | 240   | 240   | 0  | 670  | 5290  | 1787 | 987     | 3660     | 4920     | 0     |
| Set Payment Method (Check/MO) 217                      | 240   | 240   | 0  | 300  | 2760  | 904  | 518     | 2010     | 2440     | 0     |
| Set Payment Method (Check/MO) 318                      | 220   | 220   | 0  | 370  | 8180  | 2696 | 1725    | 5810     | 7590     | 0     |
| /checkout/..../success/19                              | 240   | 240   | 0  | 250  | 2830  | 931  | 583     | 1970     | 2530     | 0     |
| /checkout/..../success/ Redirect 120                   | 74    | 0     | 74 | 1030 | 8350  | 2473 | 1620    | 5590     | 7310     | 0     |

| Magento EE v1.12                                       |       |       |     |     |       |       |         |          |          |       |
|--------------------------------------------------------|-------|-------|-----|-----|-------|-------|---------|----------|----------|-------|
| Requests                                               | Total | OK    | KO  | Min | Max   | Mean  | Std Dev | 95th Pct | 99th Pct | Req/s |
| Global Information0                                    | 48044 | 47887 | 157 | 40  | 40580 | 4985  | 3290    | 10650    | 13870    | 58    |
| product page0                                          | 21327 | 21286 | 41  | 40  | 20040 | 4557  | 2820    | 9740     | 12350    | 26    |
| Click add to cart1                                     | 8340  | 8324  | 16  | 40  | 12130 | 4216  | 2217    | 7520     | 9000     | 10    |
| Click add to cart Redirect 12                          | 8324  | 8320  | 4   | 40  | 16500 | 7022  | 3449    | 11780    | 13500    | 10    |
| click checkout3                                        | 2780  | 2779  | 1   | 920 | 12350 | 6311  | 2829    | 9760     | 10950    | 3     |
| Guest Checkout start4                                  | 90    | 90    | 0   | 390 | 3690  | 2103  | 795     | 3210     | 3410     | 0     |
| Registered Checkout:Login and Pick Address5            | 2690  | 2690  | 0   | 180 | 19120 | 3298  | 4984    | 15470    | 16940    | 3     |
| Guest:Pick Billing Address 16                          | 90    | 90    | 0   | 600 | 7750  | 4348  | 1605    | 6430     | 7400     | 0     |
| Registered Checkout:Login and Pick Address Redirect 17 | 2690  | 2688  | 2   | 40  | 24360 | 5433  | 3350    | 11780    | 14060    | 3     |
| Guest:Pick Billing Address 28                          | 90    | 90    | 0   | 520 | 8240  | 3412  | 1423    | 5480     | 6810     | 0     |
| Guest:Go to pick Shipping Method9                      | 90    | 90    | 0   | 310 | 4780  | 2334  | 888     | 3300     | 4450     | 0     |
| Registered:Pick Billing Address 110                    | 90    | 90    | 0   | 370 | 10090 | 3657  | 2453    | 8660     | 9600     | 0     |
| Registered:Pick Billing Address 2-111                  | 90    | 90    | 0   | 540 | 17670 | 4148  | 2741    | 10220    | 12510    | 0     |
| Registered:Pick Billing Address 2-212                  | 90    | 90    | 0   | 530 | 12640 | 3983  | 2411    | 8290     | 11590    | 0     |
| Registered:Go to pick Shipping Method13                | 90    | 90    | 0   | 360 | 6290  | 2546  | 1401    | 5050     | 5980     | 0     |
| Set Shipping Method (Flatrate) 114                     | 180   | 180   | 0   | 450 | 7640  | 3926  | 1884    | 6550     | 7450     | 0     |
| Set Shipping Method (Flatrate): Goto Payment15         | 180   | 180   | 0   | 380 | 6290  | 2451  | 1256    | 4270     | 4830     | 0     |
| Set Payment Method (Check/MO) 116                      | 180   | 180   | 0   | 570 | 8290  | 3923  | 1996    | 6460     | 7300     | 0     |
| Set Payment Method (Check/MO) 217                      | 180   | 180   | 0   | 330 | 5230  | 2318  | 1238    | 4210     | 4810     | 0     |
| Set Payment Method (Check/MO) 318                      | 180   | 180   | 0   | 940 | 40580 | 12673 | 7957    | 25910    | 29540    | 0     |
| /checkout/./success/19                                 | 180   | 180   | 0   | 230 | 3330  | 1437  | 854     | 2870     | 3150     | 0     |
| /checkout/./success/ Redirect 120                      | 93    | 0     | 93  | 780 | 10470 | 5031  | 2251    | 7920     | 8880     | 0     |

## Estimating the Number of Visitors

Gatling does not report the number of users that cycled through during the test. However, it is possible to estimate the number using the following formula:

$$\text{Number of visitors} = \frac{\text{Concurrent sessions} \times \text{Peak load duration in seconds}}{\text{Session duration in seconds}} = \frac{1000 \times 10 \times 60}{72} = 8333.33$$

22 days ago

Did you find what you were looking for?

Yes No



- Cloud
- Setup
- Development
- Testing
- |- Functional Areas
- Tutorials

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento Community Edition (CE) Release Notes (1.8 and later)



**Note:** We'd like to make you aware that a security patch for older versions of [Magento Community Edition](#) has been posted (in the Magento Community Edition Patches section). This patch resolves a remote code execution vulnerability that enables an attacker with Magento administrator privileges to delete files and directories on a Magento installation. This vulnerability was discovered through our quarterly penetration testing process and has not been reported by merchants.

This issue was fixed in Magento Community Edition 1.8.0.0; no patch is necessary for versions 1.8.0.0 and later. Patches are available for Magento Community Edition 1.4.0.0 through 1.7.0.2. We encourage all affected merchants to apply the patch in their next regularly scheduled maintenance cycle.

Magento takes security very seriously and will continue to focus on identifying potential issues and hardening our defenses.

## Table of Contents

These Release Notes contain the following information:

- [Patches for Multiple Magento CE Versions](#)
- [Magento Community Edition \(CE\) 1.8.1.0 Release Notes](#)
- [Magento CE 1.8.0.0 Release Notes](#)

## Patches for Multiple Magento CE Versions

Magento has the following patches available for multiple Magento CE versions.

- [General Magento Connect Patches](#)

- [Magento Install Page Displays After SOAP v2 Index Page Refresh](#)
- [PHP 5.4 Patch Available](#)
- [How to get Patches for Magento CE](#)



**Note:** Some of the patches discussed in this section have EE\_1.14.0.1 in the name. These patches were all tested against CE 1.8.x as well.

## General Magento Connect Patches

*Patch name:* SUPEE-3941

- When you install a community-created translation package, the translation provided by the package overwrites any existing translations for the same items. This enables you to more easily install packages with translations.
- To improve security, Magento Connect now uses HTTPS by default to download extensions, rather than FTP.
- Extension developers can now create an extensions with a dash character in the name. Merchants can install those extensions without issues.
- Magento administrators who attempt to install an extension with insufficient file system privileges are now informed. Typically, the Magento Admin Panel runs as the web server user. If this user has insufficient privileges to the *your Magento install dir/app/code/community* directory structure, the Magento administrator sees an error message in the Magento Connect Manager.  
To set file system permissions appropriately, see [After You Install Magento: Recommended File System Ownership and Privileges](#).

## Magento Install Page Displays After SOAP v2 Index Page Refresh

*Patch name:* SUPEE-3762. Refreshing the SOAP v2 index page ([http://your-magento-host-name/index.php/api/v2\\_soap/index/](http://your-magento-host-name/index.php/api/v2_soap/index/)) results in all administrators and customers viewing the Magento installation page.

## Discover Card Validation Patch Available

Magento has fixed an issue that prevented some Discover credit cards from validating properly. The issue was that certain Discover credit card number ranges were not recognized as being valid. As a result of the fix, all Discover cards should validate properly.

The issue affects Magento CE versions 1.4.2.0–1.8.1.0.

To get a fix for the issue, see [Discover credit card validation issue: Magento EE 1.9.1.1–1.13.1.0 and CE 1.4.2.0–1.8.1.0](#).



**Important:** This is *not* a security threat. No data has been compromised or misused. It affects only the ability to validate certain credit card number ranges as valid Discover card numbers.

## PHP 5.4 Patch Available

You can use PHP 5.4 with Magento CE versions 1.6.0.0–1.8.1.0.

To get the patch, see [Getting the PHP 5.4 patch for Magento Enterprise Edition \(EE\) and Community Edition \(CE\)](#).

For more information about PHP 5.4, see [the PHP migration page](#) and [the PHP changelog](#).

# How to Get Patches For Magento CE

This section discusses how to get patches referenced in these Release Notes.

To get patches for Magento CE:

1. Log in to [www.magentocommerce.com/download](http://www.magentocommerce.com/download).
2. In the left pane, click **Downloads**.
3. Scroll down to the Magento Community Edition Patches section.
4. Follow the prompts on your screen to download a patch for your version of CE.
5. Apply the patch as discussed in [How to Apply and Revert Magento Patches](#).

## Magento Community Edition (CE) 1.8.1.0 Release Notes

See the following sections for information about changes in this release:

- [Highlights](#)
- [Security Enhancements](#)
- [Potential Issue After Upgrading to CE 1.8.1.0](#)
- [Changes in This Release](#)
- [Tax Calculation Fixes](#)
- [Fixes in Magento CE 1.8.1.0](#)

### Highlights

Magento CE 1.8.1.0 helps advance overall product quality and ease operations by providing significant tax calculation improvements, a wide range of bug fixes, and several security enhancements.

### Tax Calculation Improvements

CE 1.8.1.0 resolves Value Added Tax (VAT) and Fixed Product Tax (FPT) issues so that Magento administrators can create invoices and credit memos to give merchants access to accurate and consistent tax calculations and displays. We've also addressed:

- VAT tax calculation issues for cross-border trade
- Tax rounding issues when multiple taxes are applied
- VAT and FPT calculation issues for bundled products
- Support for the Waste Electrical and Electronic Equipment (W.E.E.E.) recycling tax in the EU

### Functional Improvements

CE 1.8.1.0 includes bug fixes across important feature areas, including the shopping cart, checkout, content management system, and product import and export function. Many of these updates came from a hackathon held with Magento community developers, which demonstrates the vitality of our development community and their powerful ability to help us advance the platform.

### Security Enhancements

CE 1.8.1.0 includes several security enhancements that were identified through our rigorous security assessment process. Magento complements its own comprehensive internal testing with quarterly

penetration testing by expert consultants and actively works with the development community to identify security issues in order to harden the platform against potential threats.

## Security Enhancements

Magento addressed the following security issues:

- Improved the password hashing algorithm.  
Magento thanks Bjorn Kraus for contributing to this fix.
- Resolved issues that could have resulted in Cross-Site Request Forgery (CSRF) in the web store.
- Resolved potential issues when issuing Return Materials Authorizations (RMAs).  
Magento thanks Ivan Chepurnyi for contributing to this fix.
- Resolved a session fixation issue when registering a user with the web store.
- Resolved issues with the expiration of file-based user sessions.
- Resolved issues that could have resulted in [Remote File Inclusion \(RFI\)](#) vulnerabilities.
- Addressed vulnerabilities in OAuth code.
- Closed a potential loophole that enables another user to possibly access personal information when viewing billing agreements.  
Magento thanks Darryl Adie and Ampersand Commerce for contributing to this fix.
- Fixed the security settings for the frontend cookie to protect user sessions.

## Issue After Upgrading to CE 1.8.1

There is a known Issue After Upgrading to CE 1.8.1 that affects you *only* if you do *not* follow the recommended procedure to upgrade to a new environment as discussed in [Getting Ready For Your Upgrade](#).

**Symptom:** After completing the upgrade, when you log in to the Admin Panel and click **System > Configuration**, a fatal error similar to the following displays in your browser:

```
Class 'Mage_Googlecheckout_Helper_Data' not found in /var/www/html/magento/app/Mage.php on line 547
```

### Solution:

1. Close the Admin Panel browser window.
2. As a user with **root** privileges, delete all files *except config.xml* from the following directory:  
`magento-install-dir/app/code/core/Mage/GoogleCheckout/etc`
3. When you log back in to the Admin Panel, everything works as expected.  
If you're still encountering errors, see [Getting Help With Your Installation or Upgrade](#).

## Changes in This Release

See the following sections for a discussion of changes in this release:

- A tax configuration option for Fixed Product Tax (FPT) has changed. This option is in the Admin Panel at **System > Configuration > SALES > Tax > Fixed Product Taxes**, option **FPT Tax Configuration**. This option replaces the **Apply Tax to FPT** option in earlier CE releases. This option specifies how FPT is calculated as follows:
  - **Not Taxed:** Click this option if your taxing jurisdiction does not tax FPT. (For example, the state of California does not tax FPT.)
  - **Taxed:** Click this option if your taxing jurisdiction does tax FPT. (For example, Canada taxes FPT.)

- **Loaded and Displayed with Tax:** Click this option if FPT is added to the order total before applying tax (for example, in EU countries).
- You can now specify a 0% tax rate. (In the Admin Panel, click **Sales > Tax > Manage Tax Zones & Rates**.).  
For more information, see the [Magento User Guide](#).
- Magento changed its recommended setting for **System > Configuration > SALES > Tax > Calculation Settings**, option **Apply Discount On Prices** as follows:
  - US and Canadian merchants: Set the value of **Apply Discount On Prices** to **Excluding Tax**.
  - EU merchants: Set the value of **Apply Discount On Prices** to **Including Tax**.
- Magento strongly recommends all merchants set **Apply Customer Tax** to **After Discount**, regardless of all other tax-related settings. This avoids issues with calculating the total product price.
- When you specify a tax rate, the **State** list is now available whenever you choose a country that has states.
- You can now specify the asterisk (\*) wildcard character for the value of **State** when you set up a new tax rate. This enables you to apply the same tax rate to all states or provinces in a particular country.
- Stores now display in the Admin Panel in **System > Manage Stores** as a three-column hierarchy, with the website in the left column, all stores associated with the website in the center column, and all store views associated with the store in the right column.  
This makes it easier for you to browse your stores and understand which websites, store views, and stores are associated with each other. The updated Manage Stores page also displays the root category for each store and the code for each website and store view.  
Magento thanks Fabrizio Branca for contributing to this fix. For more information, see [Fabrizio's blog post](#).
- For the DHL (Deprecated) shipping method to work, you must change the gateway URL as follows:
  - Log in to the Admin Panel as an administrator.
  - Click **System > Configuration > SALES > Shipping Methods**.
  - In the right pane, expand **DHL (Deprecated)**.
  - Change the value of the **Gateway URL** field to the following:
 

<http://xmlapi.dhl-usa.com/ApiLanding.asp>
  - In the upper right corner, click **Save Config**.

## Tax Calculation Fixes

Tax calculation issues can be divided into the following sections:

- [General Tax Notes](#)
- [Rounding Issues](#)
- [Display Issues](#)
- [Bundled Products Issues](#)
- [Fixed Product Tax \(FPT\) Issues](#)

## General Tax Notes

The following general fixes were made to Magento tax configuration and calculations:

- Canadian customers now receive an e-mail with the correct totals for invoices and credit

memos that include Provincial Sales Tax (PST) and Goods and Services Tax (GST).

- Resolved issues with incorrect prices and incorrect tax amounts when a custom price is used together with the configuration setting **System > Configuration > SALES > Tax > Calculation Settings**, option **Apply Tax On** set to **Original price only**.
- The tax amount is calculated correctly when:
  - The customer is in a different taxing jurisdiction than the web store
  - The configuration option **System > Configuration > SALES > Tax > Calculation Settings**, option **Catalog Prices** is set to **Including Tax**
  - The configuration option **System > Configuration > SALES > Tax > Calculation Settings**, option **Tax Calculation Method Based On** is set to **Unit Price**
- The row total including tax displayed in the shopping cart is calculated correctly when:
  - The configuration option **System > Configuration > SALES > Tax > Calculation Settings**, option **Catalog Prices** is set to **Excluding Tax**
  - The configuration option **System > Configuration > SALES > Tax > Calculation Settings**, option **Tax Calculation Method Based On** is set to **Row Total**
- The row subtotal displays the correct amount when reordering a product that includes a discount coupon.
- Multiple tax rates for a product display correctly in the Admin Panel when creating an invoice or credit memo.
- Resolved calculation errors when tax and currency conversion are applied. As a result, the price the customer views on a catalog or product page is the same as the price displayed in the shopping cart.
- A customer can now place an order when two tax rules are applied to a product, even if the tax rules specify the same tax rate.
- Resolved issues with calculating taxes on orders that are shipped to different countries that have different tax rates.
- Product prices, including taxes, display on category and product pages the same for a guest customer as for a logged-in customer. (A guest customer is a customer who does not log in to your web store; this customer belongs to the **NOT LOGGED IN** customer group.)

## Rounding Issues

The following tax rounding issues were resolved:

- Resolved a rounding issue in the tax detail display in the shopping cart when more than one tax rule is used.
- Resolved an issue reported on [stackoverflow](#) where a calculation error resulted from the following configuration:  
**System > Configuration > SALES > Tax > Calculation Settings**, option **Tax Calculation Method Based On** set to **Total**  
**System > Configuration > SALES > Tax > Calculation Settings**, option **Catalog Prices** set to **Including Tax**
- As a result of allowing a 0% tax rate, rounding errors related to different VAT jurisdictions have been resolved. For example, if a product originates in a country with VAT but is shipped to another country that has no VAT, correct prices display in the shopping cart.
- Row totals display correctly in the shopping cart when:

- A shopping cart discount is applied
- The following configuration options are set in **System > Configuration > SALES > Tax > Calculation Settings:**
  - **Catalog Prices** is set to **Including Tax**
  - **Tax Calculation Method Based On** is set to **Excluding Tax**
  - **Apply Customer Tax** is set to **After Discount**
  - **Apply Discount On Prices** is set to **Including Taxes**

## Display Issues

The following issues relate to the incorrect display of tax information in the Admin Panel or in your Magento web store:

- Shipping prices including tax display properly in the shopping cart.
- A special price now displays correctly on the product view page.
- Values displayed in PDFs for invoices and credit memos no longer overlap each other.
- Orders, invoices, and credit memos for downloadable and virtual products display the correct row total when viewed in the Admin Panel.
- Orders display the FPT in the Admin Panel when the full tax summary is specified.
- Fixed-price bundled products that include FPT now display only one price for both From and To values, regardless of how you configured the products.

## Bundled Products Issues

- The price of a dynamic bundled product is calculated correctly after being customized by the customer.
- The price of a dynamic bundled product with tiered pricing is calculated correctly after being customized by the customer.
- Resolved issues with calculating the value displayed for the price including tax for bundled products.
- The price excluding tax of a bundle product to which a discount is applied is the same:
  - When viewed on the customization page
  - after adding the bundled product to the shopping cart.
- A dynamically-priced bundled product's prices displayed for Unit Price in the shopping cart (that is, the price including tax and the price excluding tax) are now correct. Before the fix, the prices were equal, which was incorrect.
- The price of a bundled product displayed on the product view page and in the shopping cart are the same.
- The grand total including tax and the subtotal including tax displayed in the shopping cart are now identical when you specify that catalog prices include tax and the shopping cart is set to display prices with and without taxes for a dynamic bundled product that consists of two simple products.

## Fixed Product Tax (FPT) Issues

- Resolved issues in calculating FPT on a credit memo.

- With both discounts and FPT enabled (and FPT is taxable), the subtotal including tax displayed in the shopping cart is correct.
- FPT calculation for bundled products that have FPT applied to them now are now correct for all FPT configuration settings.
- The invoice total is calculated correctly for an order that has both FPT and a shopping cart discount.
- The FPT amount is now included in the Subtotal (Incl.Tax) row for partial invoice.
- Resolved an issue that resulted in FPT being applied twice to the grand total in the shopping cart.

## Fixes in Magento CE 1.8.1.0

Fixes in this release can be divided into the following categories:

- [Shopping Cart and Checkout Fixes](#)
- [Import and Export Fixes](#)
- [Shipping Fixes](#)
- [Other Fixes](#)

### Shopping Cart and Checkout Fixes

- Resolved a new customer registration issue that enabled a user to register and see another customer's dashboard.
- Resolved issues with breadcrumbs disappearing or displaying incorrectly.
- The following options are available in the Admin Panel in **System > Configuration > SALES > Sales > Gift Options**:  
**Allow Gift Wrapping on Order Level** set to **No**  
**Allow Gift Wrapping for Order Items** set to **Yes**
- Category and subcategory names display correctly. Before the issue was resolved, subcategory names that were significantly longer than the category name did not display properly.
- If a customer adds more than one product that requires products to be purchased in increments, only the products that meet the increment requirements are added. Before the fix, all products were added.
- Scheduled payments work properly.  
 Magento thanks Sylvain Raye for contributing to this fix.
- If a bundled or configurable product is out of stock, it's no longer available to check out.  
 Magento thanks Francesco Marangi for contributing to this fix.
- Placing an order in the Admin Panel correctly sets the order status to Pending.  
 Magento thanks GitHub user elframan for contributing to this fix.

### Import and Export Fixes

- Scheduled export works properly.
- You can now export a shipment to CSV after printing its shipping label.  
 Magento thanks Florinel Chis for contributing to this fix.

## Shipping Fixes

- You are not required to enter a declared value to ship with FedEx.
- FedEx shipping labels print properly; addresses are not truncated.
- Fix for the USPS change to the names of their Priority and Express shipping options in their API made on Sunday, July 28, 2013.

## Other Fixes

- Resolved issues that caused spurious errors in the Magento exception log:  
`'Zend_Date_Exception'` with message '`Invalid year, it must be between -10000 and 10000`'
- Widgets display properly on the CMS.
- The product attribute option **Use Default Value** works properly when used in a non-default store view.
- A category attribute set to store view scope displays in layered navigation.
- A store set for British Pound Sterling currency units now displays the correct currency in payment logs.
- Resolved an issue with the `Mage_Catalog_Block_Product_Abstract` class that caused errors to display on product view pages in your web store.
- Back-in-stock e-mails contain the correct content.
- You can now manage product ratings and reviews from the Admin Panel as well as from the web store.  
Magento thanks Fabian Blechschmidt Schrank for contributing to this fix.
- Resolved an issue with `Mage_Page_Block_Template_Links::addLinkBlock` to enable you to sort an array of results by position.  
Magento thanks Benjamin Marks for contributing to this fix.

# Magento Community Edition (CE) 1.8.0.0 Release Notes

See the following sections for information about changes in this release:

- [Highlights](#)
- [Security Enhancements](#)
- [United States Postal Service \(USPS\) Update](#)
- [Performance Improvements](#)
- [Tax Calculation Fixes](#)
- [API Fixes](#)
- [Fixes](#)
- [Thanks](#)

## Highlights

- Major overhaul of tax calculation formulas, correction of rounding errors, and additional assistance with configuration.
- Optimized cache adapters for single-server systems
- Upgraded Redis cache adapters for multi-server systems.  
To set up and use Redis with Magento, see [Using Redis with Magento Community Edition \(CE\) and Enterprise Edition \(EE\)](#).
- Eliminated many types of database deadlocks.

## Security Enhancements

- Errors are not displayed in a new Magento installation.
- Fixed a session fixation vulnerability in the new user registration process. Attackers can no longer abuse this flaw to take over new user accounts during registration.
- Resolved a remote code execution vulnerability that enabled an attacker to delete files and directories on the Magento installation. (The attack required access to the Admin Panel as a Magento administrator.)
- Prevent attacks that use OAuth to leak sensitive information to an attacker that knows the consumer key and user token.
- Resolved an issue that enabled attackers to gain access to billing information.  
We thank Darryl Adia (from Ampersand Commerce) for contributing to this fix.
- Resolved issues with the security of OAuth tokens and keys.
- A remote code execution vulnerability was fixed.  
We thank Bastian Ike for contributing to this fix.
- The Magento Admin Panel and web stores no longer allow web browsers to store usernames or passwords.
- The Magento web store has additional Cross Site Request Forgery (CSRF) protections, meaning an imposter can no longer impersonate a newly registered customer and perform actions on the customer's behalf.
- The cryptographic methods used to store passwords were improved to enhance security.

## United States Postal Service (USPS) Update

The USPS changed the names of their Priority and Express shipping options in their API in July 2013. To enable you to continue utilizing USPS Priority *and* Express mail methods, CE 1.8 includes a patch that addresses the issue.



**Important:** The USPS API patch has an impact on upgrading to CE 1.8 from earlier versions. If you're doing a new CE 1.8 installation, however, you don't need to do anything.

Following are details about the upgrade impact:

- Print all USPS shipping labels before upgrading; after upgrading, you will not be able to print them.
- Any shopping cart price rules that use the USPS shipping method that created before you upgrade must be re-created after you upgrade. Pre-existing USPS shipping methods do not work with shopping cart price rules after the upgrade.

# Performance Improvements

- Limited the way Magento performs large database lookups.
- Checkout performance improvements achieved by:
  - Eliminating unnecessary RSS cache cleanups when RSS functionality is disabled
  - The locale used to send a new order confirmation e-mail now first checks to see if the customer's locale is the same as the store's locale before attempting to localize the e-mail.
  - Improving the overall checkout process performance by loading the progress information for the current checkout step only
- You can load a large number of tax codes (35,000 or so) without impacting performance.

# Tax Calculation Fixes

Tax calculation issues can be divided into the following sections:

- [General Tax Notes](#)
- [Rounding Error Fixes](#)
- [Fixed Product Tax \(FPT\) Fixes](#)
- [Discount Calculation Fixes](#)
- [Display Fixes](#)

## General Tax Notes

The following general fixes were made to Magento tax configuration and calculations:

- Based on Magento testing and merchant experience, certain tax configuration settings have been determined to be susceptible to rounding issues and can be confusing to buyers. To help you avoid issues with those settings, warning messages display in the Admin Panel if you attempt to save such a configuration.  
Administrative users can choose to dismiss the messages and can still save the configuration; however, Magento strongly recommends you change the configuration in a way recommended by the details displayed in the message.  
For details, see the [Magento User Guide](#).
- Bundle pricing is more consistent as follows:
  - The calculation formula is: `Sub item price = Sub item base price * Applicable tiered price adjustment or discount, then rounded` `Bundle price = Sum (round(sub item price * qty))`
  - When non-integer quantities are multiplied by a product price, Magento rounds the resulting subtotal as follows: `round(unit price * non-integer quantity)`
- All product price information on which taxation is based are rounded to two digits of precision regardless of how many digits of precision have been loaded into the database (for example, \$10.24 instead of \$10.2385). This situation can occur when certain integrations enable third-party applications to send four-digit precision prices to Magento.  
Starting with this release those additional digits will have no impact on customer facing prices. Forcing two digits of precision enables more exact calculations involving Fixed Product Tax (FPT), discounts, and taxes—among other concerns.
- For certain Canadian provinces and localities, calculations and methods were updated to support changing legal requirements in Canada:

- Provincial Sales Tax (PST)
- Goods and Services Tax (GST)
- Taxe de vente du Québec (TVQ)—also referred to as Quebec Sales Tax (QST)

For details, see the [Magento User Guide](#).

## Rounding Error Fixes

The following issues relate to one-cent rounding errors in the web store or shopping cart:

- Calculating taxes for bundled products with tiered pricing.
- Calculating the price before customization for bundled products.
- Calculating the grand total of items added to a cart in a different order.
- Viewing an order when taxes are calculated after a discount using either row-based or unit price.
- Applying a discount to an order with a shipping address different than the billing address.
- Calculating the grand total based on the order in which products are added to the shopping cart.
- Specifying that prices display in the web store excluding tax and setting a 20% tax rate (or discount rate) now calculates the grand total correctly. It is now possible to have grand totals in amounts like 6.99, 9.99, or 99.99—regardless of the currency units used in the web store.
- Adding multiple items to a cart does not affect the accuracy with which taxation is calculated.
- Subtotal (Incl. Tax) is now correct when catalog and shipping prices include tax. Both tax and discounts are applied after tax.
- Prices displayed in the cart and on the catalog page are consistent and correct when catalog prices include tax, and when items in the catalog are set to display both including and excluding tax.

(In the Admin Panel, click **System > Configuration > SALES > Tax**. In the right pane, expand **Calculation Settings**.)

- Error in calculating the Grand Total Excl. Tax was resolved. This error occurred in a specific configuration: tax is applied to FPT, FPT is included in the subtotal, and the customer selects non-taxable flat rate shipping.

## Fixed Product Tax (FPT) Fixes

The following issues relate to errors in calculating taxes that include FPT in the web store or shopping cart:

- Price in the cart displays the correct before-tax price and grand total.
- Subtotals displayed in the cart—both Including Tax and Excluding Tax—are now correctly calculated when FPT is applied.
- Free shipping offers are now processed correctly when FPT is applied.
- FPT taxes are calculated correctly when a discount is applied.

## Discount Calculation Fixes

The following issues relate to price calculations when coupon codes or other discounts are applied in the web store or shopping cart:

- The Row Subtotal displayed in the cart is calculated correctly (that is, both Excl. Tax and Incl.

Tax are correct).

- The price for bundled items now displays with tax included if the bundle is configured to do so.
- Taxation is now correctly calculated on a product with a discounted price.
- Taxation on discounts is now calculated correctly when the ship-to country is different from the web store's default country.

## Display Fixes

The following issues relate to the incorrect display of tax information in the Admin Panel or in your Magento web store:

- Row Subtotal displays correctly in the shopping cart when:
  - FPT is applied.
  - A discount is applied to a situation where the tax the customer pays is different from the tax specified for the web store's locale (for example, when the shipping origin is different than the shipping address).
- Subtotal including tax on a credit memo is correct when one or more items in the memo includes FPT.
- Item subtotal displays correctly when a discount is applied to a purchase that includes FPT.
- If the administrator sets catalog prices to exclude tax and to display product prices in catalog as including tax, the price of the product in your web store includes applicable taxes.  
(In the Admin Panel, click **System > Configuration > SALES > Tax**. In the right pane, expand **Calculation Settings**.)
- The amount of tax displayed in the Order Totals section of the shopping cart is now correct when free shipping and a shopping cart rule discount are applied.

## API Fixes

The following are fixed in the Magento SOAP v2.0 APIs (with exceptions noted):

- Requesting a product using a call like the following returns the product with the specified numeric SKU value (8888 in the following example):  
`$result = $client->call($sessionId, 'catalog_product.info', '8888', null, null, 'sku');`
- Order status is changed correctly using `salesOrderAddComment` .
- The `shoppingCartProductMoveToCustomerQuote` method works properly.
- You can now use `from-to` complex filters to perform "window" filtration on a single field. For example, you can use `from` and `to` on the `created_at` return a list of sales orders using the `salesOrderList` .
- When you use the SOAP API v.2.0 with `WS-I Compliance` enabled to retrieve sales orders information, the server responds with the correct `Content-Length` header.
- The `productGetSpecialPrice` method returns special price information for a product, whether or not `WS-I Compliance` is enabled.
- The `shoppingCartPaymentList` method returns the list of the available payment methods for the shopping cart appropriately. The following error is no longer returned: `SOAP-ERROR: Encoding: object has no 'code' property in name`
- The following issues with `WSDL` and `WS-I Compliance` are resolved:
  - The `productAttributeAddOption` and `catalogProductAttributeUpdate` methods

are now supported when WS-I Compliance is enabled.

- The WSDL declaration for `salesOrderCreditmemoCreate` is now correct; that is, it matches the code.
- You can now add a C# web reference in Microsoft Visual Studio 2010 using the Magento WSDL.  
For example, this command no longer fails:  
`C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin> wsdl /out:Magento.cs /v http://magentohost/api/v2_soap/?wsdl`
- The **WS-I Compliance** mode WSDL now includes `catalogProductAttributeInfo`.
- **XML-RPC API:** Using the `product_custom_option.add` operation with `multicall` no longer results in redundant options.

## Fixes

Fixes in this release can be divided into the following categories:

- [Web Store and Shopping Cart Fixes](#)
- [Promotional Price Rule Fixes](#)
- [Administrative Ordering and Credit Memo Fixes](#)
- [Import Fixes](#)
- [Payment Fixes](#)
- [Other Fixes](#)

### Web Store and Shopping Cart Fixes

- A customer's account created date is correct.
- When a product price is set with website scope and an administrative user has access to only one website, the default price is taken from that website scope. Also, when saving the product on the website scope, the price is updated only in that scope and not in the default scope.
- An error no longer displays on your web store after a customer places an order. (The error message was **There has been an error processing your request. Please contact us or try again later**).
- Restricted coupon codes work properly, even if the customer has selected the **Remember me** checkbox.
- Using the Table Rates shipping option, free shipping options work properly. (In the Admin Panel, click **System > Configuration > SALES > Shipping Methods**. In the right pane, expand **Table Rates**.)
- Issues with shipping table rates have been resolved.
- Entering a value such as **10,50** (using a comma character and not a period) for **Adjustment Fee** now results in the correct amount of credit being applied to the transaction.
- Unit price for bundled products is now calculated correctly.
- The tiered price of bundled items now displays properly on the web store.
- Composite products can be successfully reordered.
- You can now use special characters in a product URL key.

- After a customer visits the sitemap, web stores URLs are no longer prepended by `/sitemap/catalog/string`.
- Welcome messages now display properly in the web store after a customer's profile information is changed.
- Recently viewed products now display updates properly.
- Armed Forces Middle East is now available for State when checking out.
- Searching for a customer's orders and returns works properly.
- Shipping is calculated correctly if you select **Using origin weight (few requests)** for **Packages Request Type**. (In the Admin Panel, click **System > Configuration > SALES > Shipping Methods > DHL (Deprecated)**.)
- Free shipping is no longer available to a customer during checkout if the option was disabled by an administrator. (In the Admin Panel, click **System > Configuration > Sales > Shipping Method > DHL(Deprecated)**, click one or more options from the **Allowed Methods** list, and, from the **Free Shipping with Minimum Order Amount** list, click **No**.)
- A user can navigate your web store while downloading a downloadable product.
- You can now specify weight units in kilograms (kg) using the FedEx shipping method.
- FedEx shipping rates are now consistent with Magento discounted rates.
- Fixed issues with United Parcel Service (UPS) shipping rates.
- UPS shipping labels have the word **SAMPLE** printed on them only when you request a sample label.
- Changes made to United States Post Office (USPS) APIs and rates have been incorporated in Magento.
- The products in a customer's wish list no longer disappear after one or more products are edited by an administrator.
- Administrators can view the contents of a customer's shopping cart.
- When a customer selects a product on your web store, the assigned category is selected in the navigation menu.

## Promotional Price Rule Fixes

The following fixes relate to administering and using shopping cart price rules and catalog price rules:

- Shopping cart price rules applied to specific customer groups work properly.
- Catalog price rules are applied properly to customer groups.
- The scope of a product attribute is now honored by a catalog price rule.
- Discounts specified by a shopping cart price rule are applied properly when a particular order is shipped to multiple addresses.
- A discount specified by a shopping cart price rule that allows for more than one use per customer is applied the correct number of times if the customer has their orders shipped to more than one address.
- When an administrative user whose role is restricted to only viewing catalog price rules, the user cannot add or edit catalog price rules.

- Shopping cart price rules now work properly with bundled products.

## Administrative Ordering and Credit Memo Fixes

- When you create an order using the Admin Panel and you have multiple stores, the **State/Province** field updates appropriately for the country in which the order is placed.
- When you create an order using the Admin Panel and you have specified a default billing address and a default shipping address, the addresses are used correctly.
- Orders placed by an administrator display in a customer's last order list.
- Product comparisons now display properly when an administrator makes a change using the Admin Panel (for example, deleting a product from a customer's comparison list).
- You can now cancel an order using the Admin Panel.
- Orders and invoices that include taxable shipping—when created in the Admin Panel—now calculate the shipping taxes properly.
- Products added to a customer's wish list by an administrator display properly.

## Import Fixes

- The quantity (QTY) of all products imports correctly.
- The value of Maximum Qty Allowed in Shopping Cart (`use_cfg_max_sale_qty`) is correct.
- The product displays correctly in layered navigation.
- Importing customer lists with capitalization variations in the e-mail address now imports the customer only once (for example, `user@example.com` and `User@example.com`).
- Issues with importing products with **Append Complex Data** selecting from a comma-separated value (.csv) file have been resolved.

## Payment Fixes

- Resolved issue sending customer e-mail when using Payflow Link.
- Security issues with Google Checkout payments have been resolved.
- Security issues with Authorize.net payments have been resolved.
- Magento conforms to the latest version of the PayPal Instant Payment Notification (IPN) guidelines.
- The contents of a shopping cart are unaffected by canceling a PayPal payment.
- Issues with not being able to continue checkout after switching payment methods have been resolved.
- You can now process partial refunds and invoices for orders that were placed using Payflow Pro.
- Payflow Link and Payments Advance now capture IPN transactions properly.
- Special characters (such as e-mail addresses) are now handled properly by the Magento Payflow API integration.
- Resolved errors with orders placed using the Website Payments Pro payment method.

- PayPal Express Checkout payments are handled properly when a shopping cart price rule is specified.
- Any PayPal Name-Value Pair (NVP) payment method no longer automatically refunds an order when a chargeback is initiated. Magento now allows the dispute to be resolved before taking the appropriate action.

PayPal NVP payment methods include: PayPal Payments Pro (including PayPal Payments Pro Hosted), Payments Standard, and all Payflow methods.

- PayPal Pro now correctly processes the shipping address for an order.
- PayPal Express Checkout and PayPal Pro now handle partial refunds properly.
- Fixed rounding errors that were preventing PayPal Express Checkout transactions from completing. The error occurred with the following configuration:

- tax calculation method based on the total
- tax calculated based on the shipping address
- catalog prices exclude tax
- shipping prices exclude tax
- customer discount applied after a discount
- discount applied to prices excluding tax
- tax applied to a custom price if available

(In the Admin Panel, click **System > Configuration > SALES > Tax**. In the right pane, expand **Calculation Settings**.)

- The order status Suspected Fraud is now supported by PayPal Payments Pro (hosted) when PayPal fraud protection is enabled. Using the Magento Admin Panel, the merchant can also accept or deny any Suspected Fraud orders and have that decision applied to the PayPal transaction.
- When sending payments in the United Kingdom, PayPal Payments Pro (hosted) now sends the value for **state** correctly. (Before the fix, **city** was sent as the value for **state**.)
- Using the Ogone payment method, transactions display in the Magento Admin Panel after you capture them.
- When an administrator places an order and uses PSi Gate, then cancels the order, the PSi Gate gateway displays both the order and the void transactions.
- The following fields related to PayPal's Payflow Pro Gateway payment method are now implemented properly:
  - **[custref]** is the Magento customer's ID
  - **[INVNUM]** is Magento's order number
- Fixed spurious **Gateway error: Void error: V18A4B18E0F9** has been captured errors when canceling partially invoiced orders when the Payflow Pro processor was used to process the payment.
- **3-D secure fix that affects UK merchants only:** 3-D Secure for UK merchants implementing Direct Payment works properly.
- Partial captures are now supported for the following PayPal payment methods: Express Checkout, Payments Pro Payflow Edition, and PayPal Standard.
- Using the PayPal Express Checkout method, a recently added customer can check out without

the error This customer email already exists.

## Other Fixes

- Issues related to the DHL shipping method for picking up and pricing orders on holidays have been resolved as follows:
  - If the current date is a weekend, Magento chooses next Monday as the pick-up date.
  - If the current date is a holiday, Magento requests from DHL information about the next five consecutive days to find a workday on which to pick up the order.
  - If there is no workday in the five consecutive days following a holiday, the DHL shipping method is unavailable.
- The `.htaccess.sample` provided with Magento now includes `php_value memory_limit 512M` to be consistent with the [Magento system requirements](#).
- You can now install or upgrade to CE 1.8.0.0 if your Magento database had a table prefix (for example, all tables start with `mage_` because you specified a tables prefix during installation).
- MySQL database deadlock issues were resolved.
- CE 1.8 is now [World Wide Web Consortium \(W3C\) compliant](#).
- When an administrative user whose role is restricted to managing products attempts to edit Inventory settings (**Catalog > Manage Products, Inventory**), only the available options display.
- Related product information updates appropriately in the Admin Panel.
- Issues with editing product inventory settings and category attributes using the Google Chrome web browser have been resolved.
- Rolling back after a backup now works properly. (The Magento backup and rollback options are available in the Admin Panel in **System > Tools > Backup**.)
- You can now fetch data for a PayPal Settlement Report using a custom Secure FTP (SFTP) server.
- You can now save a category with the option **Available Product Listing Sort By: Best value or Price** enabled.

## Thanks

Magento acknowledges and thanks everyone in the Magento Community who contributed to this release, including Colin Mollenhour for Redis modules.

 22 days ago



## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento Enterprise Edition (EE) Release Notes (1.13 and later)

## Table of Contents

These Release Notes contain the following information:

- [Patches for Multiple Magento EE Versions](#)
- [Magento EE 1.13.1.0 Release Notes](#)
- [Magento EE 1.13.0.2 Release Notes](#)
- [Magento EE 1.13.0.0 Release Notes](#)

## Patches for Multiple Magento EE Versions

Magento has the following patches available for multiple Magento EE versions.

- [General Magento Connect Patches](#)
- [Magento Install Page Displays After SOAP v2 Index Page Refresh](#)
- [Multiple Simultaneous Magento Administrators](#)
- [Discover Card Validation Patch Available](#)
- [PHP 5.4 Patch Available](#)
- [How to get Patches for Magento EE](#)



**Note:** Some of the patches discussed in this section have **EE\_1.14.0.1** in the name. These patches were all tested against EE 1.13.x as well.

## General Magento Connect Patches

*Patch name:* PATCH\_SUPEE-3941\_EE\_1.14.0.1\_v1-2014-08-12-12-10-06.sh

- When you install a community-created translation package, the translation provided by the package overwrites any existing translations for the same items. This enables you to more easily install packages with translations.
- To improve security, Magento Connect now uses HTTPS by default to download extensions, rather than FTP.
- Extension developers can now create an extensions with a dash character in the name. Merchants can install those extensions without issues.
- Magento administrators who attempt to install an extension with insufficient file system privileges are now informed. Typically, the Magento Admin Panel runs as the web server user. If this user has insufficient privileges to the *your Magento install dir/app/code/community* directory structure, the Magento administrator sees an error message in the Magento Connect Manager.  
To set file system permissions appropriately, see [After You Install Magento: Recommended File System Ownership and Privileges](#).

## Magento Install Page Displays After SOAP v2 Index Page Refresh

*Patch name:* PATCH\_SUPEE-3762\_EE\_1.14.0.1\_v1.sh. Refreshing the SOAP v2 index page ([http://your-magento-host-name/index.php/api/v2\\_soap/index/](http://your-magento-host-name/index.php/api/v2_soap/index/)) results in all administrators and customers viewing the Magento installation page.

## Multiple Simultaneous Magento Administrators

*Patch name:* PATCH\_SUPEE-3819\_EE\_1.14.0.1\_v1.sh. Multiple Magento administrators can simultaneously add new products; or edit descriptions, edit prices, or edit stock quantities of existing products without causing deadlocks, key violations, or critical data errors. Together with applying the patch, you must set all indexers to Update when scheduled as follows:

- Log in to the Magento Admin Panel as an administrator.
- Click **System > Configuration**.
- In the left navigation bar, from the ADVANCED group, click **Index Management**.
- Expand **Indexing Options**.
- From each list, click **Update when scheduled**.
- Click **Save Config** in the upper right corner of the page.

## Discover Card Validation Patch Available

Magento has fixed an issue that prevented some Discover credit cards from validating properly. The issue was that certain Discover credit card number ranges were not recognized as being valid. As a result of the fix, all Discover cards should validate properly.

The issue affects EE versions 1.9.1.1 through 1.13.1.0.

To get a fix for the issue, see [Discover credit card validation issue: Magento EE 1.9.1.1–1.13.1.0 and CE 1.4.2.0–1.8.1.0](#).



**Important:** This is *not* a security threat. No data has been compromised or misused. It affects only the ability to validate certain credit card number ranges as valid Discover card numbers.

## PHP 5.4 Patch Available

You can use PHP 5.4 with Magento EE versions 11.0.0.0–1.13.1.0.

To get the patch, see [Getting the PHP 5.4 patch for Magento Enterprise Edition \(EE\) and Community Edition \(CE\)](#).

For more information about PHP 5.4, see [the PHP migration page](#) and [the PHP changelog](#).

## How to get Patches for Magento EE

This section discusses how to get patches referenced in these Release Notes. Magento has other patches available from the EE support portal and the [partner portal](#); you can use the following instructions to install any of those patches as well.

To get patches for Magento EE:

1. Log in to [www.magentocommerce.com](http://www.magentocommerce.com).
2. In the left pane, click **Downloads**.
3. In the right pane, click **Magento Enterprise Edition**.
4. Follow the prompts on your screen to download a patch for your version of EE.
5. Apply the patch as discussed in [How to Apply and Revert Magento Patches](#).

## Magento EE 1.13.1.0 Release Notes

See the following sections for information about changes in this release:

- [Highlights](#)
- [Security Enhancements](#)
- [Potential Issue After Upgrading to EE 1.13.1.0](#)
- [Changes in This Release](#)
- [Tax Calculation Fixes](#)
- [Fixes in Magento EE 1.13.1.0](#)

## Highlights

Magento EE 1.13.1.0 helps advance overall product quality and ease operations by providing significant tax calculation improvements, a wide range of bug fixes, and several security enhancements.

## Tax Calculation Improvements

EE 1.13.1.0 resolves Value Added Tax (VAT) and Fixed Product Tax (FPT) issues so that Magento administrators can create invoices and credit memos to give merchants access to accurate and consistent tax calculations and displays. We've also addressed:

- VAT tax calculation issues for cross-border trade
- Tax rounding issues when multiple taxes are applied
- VAT and FPT calculation issues for bundled products
- Support for the Waste Electrical and Electronic Equipment (W.E.E.E.) recycling tax in the EU

## Functional Improvements

EE 1.13.1.0 includes bug fixes across important feature areas, including the shopping cart, checkout, content management system, and product import and export function. Many of these updates came from a hackathon held with Magento community developers, which demonstrates the

vitality of our development community and their powerful ability to help us advance the platform.

## Security Enhancements

EE 1.13.1.0 includes several security enhancements that were identified through our rigorous security assessment process. Magento complements its own comprehensive internal testing with quarterly penetration testing by expert consultants and actively works with the development community to identify security issues in order to harden the platform against potential threats.

## Security Enhancements

Magento addressed the following security issues:

- Improved the password hashing algorithm.  
Magento thanks Bjorn Kraus for contributing to this fix.
- Resolved issues that could have resulted in Cross-Site Request Forgery (CSRF) in the web store.
- Resolved potential issues when issuing Return Materials Authorizations (RMAs).  
Magento thanks Ivan Chepurnyi for contributing to this fix.
- Resolved a session fixation issue when registering a user with the web store.
- Resolved a [cross-site scripting \(XSS\) issue reported in EE 1.13.0.0](#).  
Magento thanks Myke Hines, Fox, and Optero for contributing to this fix.
- Resolved issues with the expiration of file-based user sessions.
- Resolved issues that could have resulted in [Remote File Inclusion \(RFI\)](#) vulnerabilities.
- Addressed vulnerabilities in OAuth code.
- Closed a potential loophole that enables another user to possibly access personal information when viewing billing agreements.  
Magento thanks Darryl Adie and Ampersand Commerce for contributing to this fix.
- Resolved a remote code execution vulnerability that enabled an attacker to delete files and directories on the Magento installation. (The attack required access to the Admin Panel as a Magento administrator.)
- Fixed the security settings for the `frontend` cookie to protect user sessions.

## Potential Issue After Upgrading to EE 1.13.1.0

There is a known issue after upgrading to EE 1.13.1 that affects you *only* if you do *not* follow the recommended procedure to upgrade to a new environment as discussed in [Getting Ready For Your Upgrade](#).

**Symptom:** After completing the upgrade, when you log in to the Admin Panel and click **System > Configuration**, a fatal error similar to the following displays in your browser:

```
Class 'Mage_Googlecheckout_Helper_Data' not found in /var/www/html/magento/app/Mage.php on line 547
```

### Solution:

1. Close the Admin Panel browser window.
2. As a user with `root` privileges, delete all files `except config.xml` from the following directory:

```
magento-install-dir/app/code/core/Mage/GoogleCheckout/etc
```

- When you log back in to the Admin Panel, everything works as expected.  
If you're still encountering errors, see [Getting Help With Your Installation or Upgrade](#).

## Changes in This Release

See the following sections for a discussion of changes in this release:

- EE's Payment Bridge module has been updated to the latest version.  
For more information, see [this Magento blog post](#).
- A tax configuration option for Fixed Product Tax (FPT) has changed. This option is in the Admin Panel at **System > Configuration > SALES > Tax > Fixed Product Taxes**, option **FPT Tax Configuration**. This option replaces the **Apply Tax to FPT** option in earlier EE releases. This option specifies how FPT is calculated as follows:
  - Not Taxed**: Click this option if your taxing jurisdiction does not tax FPT. (For example, the state of California does not tax FPT.)
  - Taxed**: Click this option if your taxing jurisdiction does tax FPT. (For example, Canada taxes FPT.)
  - Loaded and Displayed with Tax**: Click this option if FPT is added to the order total before applying tax (for example, in EU countries).
- You can now specify a 0% tax rate. (In the Admin Panel, click **Sales > Tax > Manage Tax Zones & Rates**).  
For more information, see the [Magento User Guide](#).
- Magento changed its recommended setting for **System > Configuration > SALES > Tax > Calculation Settings**, option **Apply Discount On Prices** as follows:
  - US and Canadian merchants: Set the value of **Apply Discount On Prices** to **Excluding Tax**.
  - EU merchants: Set the value of **Apply Discount On Prices** to **Including Tax**.
- Magento strongly recommends all merchants set **Apply Customer Tax to After Discount**, regardless of all other tax-related settings. This avoids issues with calculating the total product price.
- When you specify a tax rate, the **State** list is now available whenever you choose a country that has states.
- You can now specify the asterisk (\*) wildcard character for the value of **State** when you set up a new tax rate. This enables you to apply the same tax rate to all states or provinces in a particular country.
- Stores now display in the Admin Panel in **System > Manage Stores** as a three-column hierarchy, with the website in the left column, all stores associated with the website in the center column, and all store views associated with the store in the right column.  
This makes it easier for you to browse your stores and understand which websites, store views, and stores are associated with each other. The updated Manage Stores page also displays the root category for each store and the code for each website and store view.  
Magento thanks Fabrizio Branca for contributing to this fix. For more information, see [Fabrizio's blog post](#).
- For the DHL (Deprecated) shipping method to work, you must change the gateway URL as follows:
  - Log in to the Admin Panel as an administrator.
  - Click **System > Configuration > SALES > Shipping Methods**.

c. In the right pane, expand **DHL (Deprecated)**.

d. Change the value of the **Gateway URL** field to the following:

`http://xmlapi.dhl-usa.com/ApiLanding.asp`

e. In the upper right corner, click **Save Config**.

## Tax Calculation Fixes

Tax calculation issues can be divided into the following sections:

- [General Tax Notes](#)
- [Rounding Issues](#)
- [Display Issues](#)
- [Bundled Products Issues](#)
- [Fixed Product Tax \(FPT\) Issues](#)

### General Tax Notes

The following general fixes were made to Magento tax configuration and calculations:

- Canadian customers now receive an e-mail with the correct totals for invoices and credit memos that include Provincial Sales Tax (PST) and Goods and Services Tax (GST).
- Resolved issues with incorrect prices and incorrect tax amounts when a custom price is used together with the configuration setting **System > Configuration > SALES > Tax > Calculation Settings**, option **Apply Tax On** set to **Original price only**.
- The tax amount is calculated correctly when:
  - The customer is in a different taxing jurisdiction than the web store
  - The configuration option **System > Configuration > SALES > Tax > Calculation Settings**, option **Catalog Prices** is set to **Including Tax**
  - The configuration option **System > Configuration > SALES > Tax > Calculation Settings**, option **Tax Calculation Method Based On** is set to **Unit Price**
- The row total including tax displayed in the shopping cart is calculated correctly when:
  - The configuration option **System > Configuration > SALES > Tax > Calculation Settings**, option **Catalog Prices** is set to **Excluding Tax**
  - The configuration option **System > Configuration > SALES > Tax > Calculation Settings**, option **Tax Calculation Method Based On** is set to **Row Total**
- The row subtotal displays the correct amount when reordering a product that includes a discount coupon.
- Multiple tax rates for a product display correctly in the Admin Panel when creating an invoice or credit memo.
- Resolved calculation errors when tax and currency conversion are applied. As a result, the price the customer views on a catalog or product page is the same as the price displayed in the shopping cart.
- A customer can now place an order when two tax rules are applied to a product, even if the tax rules specify the same tax rate.

- Resolved issues with calculating taxes on orders that are shipped to different countries that have different tax rates.
- Product prices, including taxes, display on category and product pages the same for a guest customer as for a logged-in customer. (A guest customer is a customer who does not log in to your web store; this customer belongs to the **NOT LOGGED IN** customer group.)

## Rounding Issues

The following tax rounding issues were resolved:

- Resolved a rounding issue in the tax detail display in the shopping cart when more than one tax rule is used.
- Resolved an issue reported on [stackoverflow](#) where a calculation error resulted from the following configuration:  
**System > Configuration > SALES > Tax > Calculation Settings**, option **Tax Calculation Method Based On** set to **Total**  
**System > Configuration > SALES > Tax > Calculation Settings**, option **Catalog Prices** set to **Including Tax**
- As a result of allowing a 0% tax rate, rounding errors related to different VAT jurisdictions have been resolved. For example, if a product originates in a country with VAT but is shipped to another country that has no VAT, correct prices display in the shopping cart.
- Row totals display correctly in the shopping cart when:
  - A shopping cart discount is applied
  - The following configuration options are set in **System > Configuration > SALES > Tax > Calculation Settings**:
    - **Catalog Prices** is set to **Including Tax**
    - **Tax Calculation Method Based On** is set to **Excluding Tax**
    - **Apply Customer Tax** is set to **After Discount**
    - **Apply Discount On Prices** is set to **Including Taxes**

## Display Issues

The following issues relate to the incorrect display of tax information in the Admin Panel or in your Magento web store:

- Shipping prices including tax display properly in the shopping cart.
- A special price now displays correctly on the product view page.
- Values displayed in PDFs for invoices and credit memos no longer overlap each other.
- Orders, invoices, and credit memos for downloadable and virtual products display the correct row total when viewed in the Admin Panel.
- Orders display the FPT in the Admin Panel when the full tax summary is specified.
- Fixed-price bundled products that include FPT now display only one price for both From and To values, regardless of how you configured the products.

## Bundled Products Issues

- The price of a dynamic bundled product is calculated correctly after being customized by the customer.

- The price of a dynamic bundled product with tiered pricing is calculated correctly after being customized by the customer.
- Resolved issues with calculating the value displayed for the price including tax for bundled products.
- The price excluding tax of a bundle product to which a discount is applied is the same:
  - When viewed on the customization page
  - after adding the bundled product to the shopping cart.
- A dynamically-priced bundled product's prices displayed for Unit Price in the shopping cart (that is, the price including tax and the price excluding tax) are now correct. Before the fix, the prices were equal, which was incorrect.
- The price of a bundled product displayed on the product view page and in the shopping cart are the same.
- The grand total including tax and the subtotal including tax displayed in the shopping cart are now identical when you specify that catalog prices include tax and the shopping cart is set to display prices with and without taxes for a dynamic bundled product that consists of two simple products.

## Fixed Product Tax (FPT) Issues

- Resolved issues in calculating FPT on a credit memo.
- With both discounts and FPT enabled (and FPT is taxable), the subtotal including tax displayed in the shopping cart is correct.
- FPT calculation for bundled products that have FPT applied to them now are now correct for all FPT configuration settings.
- The invoice total is calculated correctly for an order that has both FPT and a shopping cart discount.
- The FPT amount is now included in the Subtotal (Incl.Tax) row for partial invoice.
- Resolved an issue that resulted in FPT being applied twice to the grand total in the shopping cart.

## Fixes in Magento EE 1.13.1.0

Fixes in this release can be divided into the following categories:

- [Shopping Cart and Checkout Fixes](#)
- [Import and Export Fixes](#)
- [Shipping Fixes](#)
- [Payment Fixes](#)
- [Other Fixes](#)

## Shopping Cart and Checkout Fixes

- Abandoned cart e-mails are sent at the scheduled time.
- Resolved a new customer registration issue that enabled a user to register and see another customer's dashboard.

- Resolved issues with breadcrumbs disappearing or displaying incorrectly.
- With the following configuration, gift pricing is applied properly for more than one item. The following options are available in the Admin Panel in **System > Configuration > SALES > Sales > Gift Options**:
  - Allow Gift Wrapping on Order Level** set to **No**
  - Allow Gift Wrapping for Order Items** set to **Yes**
- Category and subcategory names display correctly. Before the issue was resolved, subcategory names that were significantly longer than the category name did not display properly.
- If a customer adds more than one product that requires products to be purchased in increments, only the products that meet the increment requirements are added. Before the fix, all products were added.
- Scheduled payments work properly.  
Magento thanks Sylvain Raye for contributing to this fix.
- If a bundled or configurable product is out of stock, it's no longer available to check out.  
Magento thanks Francesco Marangi for contributing to this fix.
- Placing an order in the Admin Panel correctly sets the order status to Pending.  
Magento thanks GitHub user elframan for contributing to this fix.

## Import and Export Fixes

- Scheduled export works properly.
- You can now export a shipment to CSV after printing its shipping label.  
Magento thanks Florinel Chis for contributing to this fix.

## Shipping Fixes

- You are not required to enter a declared value to ship with FedEx.
- FedEx shipping labels print properly; addresses are not truncated.
- Fix for the USPS change to the names of their Priority and Express shipping options in their API made on Sunday, July 28, 2013.

## Payment Fixes

- The PayFlow Pro payment method now allows line items with a negative value.
- You can now use the GSI Payment Service with Magento Payment Bridge.
- Made the following fixes to eWay Direct:
  - Updated the Payment Bridge console for eWay Direct.
  - You can now process credit memos for eWay Direct.
  - Capture for eWay direct now works as expected (that is, after the order is placed, the transaction is *not* captured but *can be* refunded).
- ( Because refunds are not supported by the Paybox Direct method, you cannot process refunds using the Admin Panel.
- After creating a refund for a First Data transaction, the transaction is closed.
- You can now pay for an order using Sagepay.

- You can now pay for an order using Worldpay.
- Line item details are now available for orders placed using Sage Pay.
- Authorize.net sends only one validation request per transaction with Customer Information Management (CIM) enabled.

## Other Fixes

- Resolved issues that caused spurious errors in the Magento exception log: `'Zend_Date_Exception'` with message `'Invalid year, it must be between -10000 and 10000'`
- Merging CMS pages no longer results in errors.
- Widgets display properly on the CMS.
- Previewing a CMS page works properly.
- The product attribute option **Use Default Value** works properly when used in a non-default store view.
- A category attribute set to store view scope displays in layered navigation.
- A store set for British Pound Sterling currency units now displays the correct currency in payment logs.
- Resolved an issue with the `Mage_Catalog_Block_Product_Abstract` class that caused errors to display on product view pages in your web store.
- Fixed issues with customer segments.
- Back-in-stock e-mails contain the correct content.
- You can now manage product ratings and reviews from the Admin Panel as well as from the web store.  
Magento thanks Fabian Blechschmidt Schrank for contributing to this fix.
- Resolved an issue with `Mage_Page_Block_Template_Links::addLinkBlock` to enable you to sort an array of results by position.  
Magento thanks Benjamin Marks for contributing to this fix.

## Magento EE 1.13.0.2 Release Notes

In response to customer feedback about EE 1.13.0.0 and EE 1.13.0.1, Magento has modified the functionality to smooth the migration path from earlier EE versions and support duplicate category URL keys.

See the following sections for a discussion of changes in this release:

- [Search Engine Optimization in Magento EE 1.13.0.2](#)
- [Known Issue in EE 1.13.0.2](#)
- [Patch Available for EE 1.13.0.2](#)
- [Changes in EE 1.13.0.2](#)
- [Fixes in Magento EE 1.13.0.2](#)

### **Important:**

- Perform *all* new installations and upgrades to Magento EE 1.13.0.2—*not to Magento EE*

[1.13.0.0 or 1.13.0.1](#)—to avoid issues with missing products on your web store due to duplicate URL key issues during the upgrade.

- The upgrade to EE 1.13.0.2 involves tasks not required for other EE upgrades. For more information, see [Important Information About Upgrading to Magento Enterprise Edition \(EE\) 1.13.0.2](#).

## Search Engine Optimization in Magento EE 1.13.0.2

This section discusses how search engine optimization (SEO) works in EE 1.13.0.2. This section is *not* intended to be a tutorial on SEO.

See one of the following for more information:

- [Comparing EE 1.13.0.2 with EE 1.12.0.2](#)
- [URL Key Uniqueness Rules in Magento EE 1.13.0.2](#)
- [URL Key Examples](#)
- [Prioritizing URL Resolution](#)
- [For More Information about SEO](#)

## Comparing EE 1.13.0.2 with EE 1.12.0.2

The following sections quickly summarize the changes you'll see in EE 1.13.0.2:

- [Product URL key uniqueness](#)
- [No more chained redirects](#)
- [Per-entity indexing](#)

### Product URL key uniqueness

#### **Description:**

The main difference introduced in EE 1.13.0.2 is that product URL keys must be globally unique among all websites, stores, and views. You can no longer have two different products that have the same URL key.

#### **Benefit:**

A single URL leads uniquely to a product page.

#### **Discussion:**

In EE 1.12, it was possible to have multiple products with the same URL key; however, every time the indexer ran, it silently assigned a numerical suffix to duplicates (for example, `shoes` became `shoes-1` and so on).

Every time this happened, another URL rewrite was created, resulted in a set of *chained redirects* for the same product. Having multiple URLs for a product dilutes the effectiveness of URL in search engine weightings, especially if you enabled canonical URLs. (As discussed in [this article on Google's blog](#), a *canonical URL* is a public specification of your preferred URL. The canonical URL is used by any search engine when crawling and indexing your site.)

This behavior was not clear to merchants and had the effect of diluting search engine weightings.

In EE 1.13.0.2, there is a single, unique way to access a product (or multiple ways if you use the category path in URLs).

## No more chained redirects

### Description:

Magento addressed the indexer issue that resulted in suffixes being silently added to products with duplicate URL keys. In EE 1.13.0.2, duplicate URL keys are not allowed.

### Benefit:

Search engines recognize the canonical URL, which improves the product's weighting in search results. (All of the weighting goes to the canonical URL.)

## Per-entity indexing

### Description:

EE 1.13.0.2 uses per-entity indexing that indexes custom URL redirects, categories, and products—as opposed to a global indexer.

### Benefit:

Similar to chained redirects, in EE 1.12, if a product had the same URL key as its parent category, the indexer assigned an incrementing numeric suffix to either the category or the product. This was done without the merchant's knowledge and was confusing as well.

### Discussion:

In EE 1.12, if you named a top-level category `slippers` and had product also named `slippers`, the indexer allowed to access to the category using a URL like the following:

`http://www.example.com/slippers-1`

In EE 1.13.0.2, the same product can be accessed using a URL like:

`http://www.example.com/slippers`

There is a new Admin Panel setting to specify how indexing should be prioritized. This setting, **System > Configuration > CATALOG > Catalog > Search Engine Optimizations > Priority for Duplicated URL Keys**, is discussed in more detail in [Prioritizing URL Resolution](#).

## URL Key Uniqueness Rules in Magento EE 1.13.0.2

The following entities can be indexed and therefore have a requirement for URL key uniqueness:

- Categories
- Products (including custom URL redirects)
- Content Management System (CMS)

Uniqueness rules for each entity type follow:

| ENTITY TYPE                                          | UNIQUENESS RULE                                                                                      |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| Product, including custom URL redirects <sup>†</sup> | All product URL keys must be globally unique.                                                        |
| Category                                             | Category URL keys must be unique only in the same level in the hierarchy; for example <i>website</i> |

|     |                                                                                                                                                                                                                                                                                                           |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | <p><i>root category</i><br/> <i>store view</i><br/> <i>category tree</i><br/> <i>*category name*</i></p> <p><b>Note:</b> Uniqueness rules apply to inactive categories as well. You cannot use the same URL key for both an active and inactive category at the same level in the category hierarchy.</p> |
| CMS | CMS URL keys, like category URL keys, must be unique only in the same level in the hierarchy.                                                                                                                                                                                                             |

†—*Custom URL redirect* refers to a product's **Create Custom Redirect for old URL** option.

## URL Key Examples

The following table shows category URL keys that are allowed. (The URL key is **shoes** for all entities in the table.)

| URL EXAMPLES                                                                                                 | REASON ALLOWED                  |
|--------------------------------------------------------------------------------------------------------------|---------------------------------|
| <code>http://example.com/mens/shoes</code><br><code>http://example.com/womens/shoes</code>                   | Different category hierarchies. |
| <code>http://example1.com/shoes</code><br><code>http://example2.com/shoes</code>                             | Different domains.              |
| <code>http://example.com/shoes</code> (store view 1)<br><code>http://example.com/shoes</code> (store view 2) | Different store views.          |

### Notes:

- You *cannot* have the same category URL key for two categories at the same level in the same store view.
- You can optionally add the store code to the URL path (in the Admin Panel, click **System > Configuration > GENERAL > Web, Add Store Code to URLs**).

## Prioritizing URL Resolution

Suppose you have the following set of URL keys. All of them are allowed because they're for different entity types.

| ENTITY TYPE          | ENTITY NAME | URL KEY | SAMPLE URL                                     |
|----------------------|-------------|---------|------------------------------------------------|
| Category             | shoes       | shoes   | <code>http://www.example.com/shoes.html</code> |
| Product              | shoes       | shoes   | <code>http://www.example.com/shoes.html</code> |
| Custom URL redirects | shoes       | shoes   | <code>http://www.example.com/shoes.html</code> |

**Question:** What happens when a web store visitor requests `http://www.example.com/shoes.html`?

**Answer:** You control the response in the Admin Panel. Click **System > Configuration > CATALOG > Search Engine Optimizations**. In the right pane, click an option from the **Priority for Duplicated**

**URL Keys** list. Some examples follow:

| PRIORITY SETTING                                                          | RESULT              |
|---------------------------------------------------------------------------|---------------------|
| Default setting:<br>1. Redirect <sup>†</sup><br>2. Category<br>3. Product | Custom URL redirect |
| 1. Category<br>2. Redirect<br>3. Product                                  | Shoes category      |

†—*Custom URL redirect* refers to a product's **Create Custom Redirect for old URL** option.

The other options are:

- Redirect - Product - Category
- Category - Product - Redirect
- Product - Redirect - Category
- Product - Category - Redirect

In the event no URL key matches your priority setting, Magento continues through the priorities in order until a match is found.



**Note:** CMS URL keys are always prioritized last.

## For More Information about SEO

For more information about SEO, see:

- [Magento SEO](#)
- [Magento SEO forum](#)
- [SearchEngineLand—What is Search Engine Optimization?](#)
- [Google—Search Engine Optimization Guide \(PDF\)](#)
- [Google—Demystifying the "duplicate content penalty"](#)

## Known Issue in EE 1.13.0.2

After changing the value for product or category suffix, previous suffixes do not work.

For example, if a category URL suffix was set to **.html** and you change it to **.php**, categories that had been using the .html suffix display an HTTP 404 (Not Found) error in your web store.

(In the Admin Panel, click **System > Configuration > CATALOG > Catalog > Search Engine Optimizations**. In the right pane, the options are named **Product URL Suffix** and **Category URL Suffix**.)

## Patch Available for EE 1.13.0.2

The United States Postal Service (USPS) changed the names of their Priority and Express shipping options in their API on Sunday, July 28, 2013. Magento has a patch available; however, this patch is not included in new EE 1.13.0.2 installations.



**Note:** If you applied the patch before you upgraded to EE 1.13.0.2, you don't have to do anything.

For new EE 1.13.0.2 installations to continue utilizing USPS Priority *and* Express mail methods, you must install the patch we've created to address the issue.

Get the patch from the EE support portal by logging in to [magentocommerce.com](http://magentocommerce.com).

## Changes in EE 1.13.0.2

- You now have the option of specifying a store view when creating a URL redirect. There is also a Store column on the **Catalog > URL Redirects** page.  
This column specifies the store view for which the URL redirect is defined.
- Root categories have no **URL Key** field. The root category URL key was never used, so the field was eliminated.
- Validation for the **URL key** field was improved. URL keys can only contain alphanumeric characters (a-z, 0-9) and the hyphen or dash character (-).
- Categories and products in categories display with the full category path if the following setting is made in the Admin Panel: **System > Configuration > CATALOG > Catalog**. Set the option named **Use Categories Path for Product URLs** to **Yes**.
- When setting up a product URL redirect, the name of the button on the category selection page has changed from **Skip choosing category** to **Skip Category Selection**.
- There is a new setting in the Admin Panel to set the priority for resolving duplicate URLs for different entities (that is, custom URL redirects, categories, and products). For more information, see [Prioritizing URL Resolution](#).
- URL key uniqueness rules apply to inactive categories as well. You cannot use the same URL key for both an active and inactive category at the same level in the category hierarchy.
- The Google sitemap for products and categories is updated with new or updated URL keys at the interval you specify in the Admin Panel: **System > Configuration > CATALOG > Google Sitemap**. In the right pane, expand **Generation Settings**.
- The following apply to new products when you do not explicitly set a value for the **URL key** field:
  - Duplicating a results in a URL key with an appended index number. For example, if you duplicate a product with a URL key of `testurlkey`, the new product's URL key might be `testurlkey-1`.
  - If you create a new product with a name that matches an existing URL key, the new product's URL key has the product ID appended to it. For example, if there is currently a product with a URL of `product1` and you create a new product with the name `Product1` and the product has an ID of 500, the new product's URL key is `product1-500`.



**Note:** To avoid having Magento specify a URL key for you, you can enter one yourself. Make sure the URL key you specify is unique among all products, including products across store views.

- The following apply to enabling the canonical meta tag options for categories and products:
  - If **Use Canonical Meta Tag for Categories** is enabled, the category page on your web

store includes a canonical URL to the full category URL. For example,  
`http://www.example.com/mens/shoes`

- If **Use Canonical Meta Tag for Products** is enabled, the product page includes a canonical URL to `domain-name/product-url-key` because product URL keys must be globally unique.

If you also enable the option **Use Categories Path for Product URLs**, the canonical URL is still `domain-name/product-url-key` but the product can also be accessed using its full URL (including the category hierarchy). Examples:

If the product URL key is `producturlkey` and it's assigned to the **Apparel > Womens > Purses** category, the product can be accessed using both of the following URLs:

`http://www.example.com/producturlkey`

`http://www.example.com/apparel/womens/purses/producturlkey`

The *canonical URL* for the product is `http://www.example.com/producturlkey`.

**Notes:**

- The canonical URL options are available in the Admin Panel by clicking **System > Configuration > CATALOG > Catalog**. In the right pane, expand **Search Engine Optimizations**. The options are named **Use Canonical Link Meta Tag For Categories** and **Use Canonical Link Meta Tag For Products**.
- The option to add the category path to a product URL is available in the same location in the Admin Panel. The option name is **Use Categories Path for Product URLs**.

## Fixes in Magento EE 1.13.0.2

This section discusses fixes made in EE 1.13.0.2.

- Resolved a critical database deadlock in the URL rewrite indexer. The deadlock resulted in the following message in the exception log: `SQLSTATE[40001]: Serialization failure: 1213 Deadlock found when trying to get lock`.
- The following layered navigation issues were addressed:
  - If a subcategory contains a product that is associated with the parent category, you can view the subcategory in your web store.
  - Clicking a link for a product attribute in an anchor category works.
- You can now install or upgrade to EE 1.13.0.2 if your Magento database had a table prefix (for example, all tables start with `mage_` because you specified a tables prefix during installation).
- Resolved the following upgrade issues:
  - After upgrading from an earlier version like EE 1.12.0.2, errors such as the following no longer display in the Magento exception log: `SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry`
  - Resolved an issue that prevented web store users from viewing products that had duplicate URL keys before upgrading.
- Resolved the following issues with URL keys in store views:
  - Changing the store view no longer results in product or category 404 (Not Found) errors.
  - Resolved an issue that categories created from a store view had the wrong URL format in the web store. Examples follow:  
*Incorrect* (before the fix): `/catalog/category/view/s/newcategory/id/36/`  
*Correct*: `/catalog/newcategory/`

- Resolved an issue where a product could be viewed using a category to which it was not assigned.

This behavior was associated with the following Admin Panel setting: **System > Configuration > CATALOG > Catalog**.

- Moving a category no longer results in HTTP 404 (Not Found) errors viewing a category's product in the web store.
- Categories display in the web store with the correct URL key, even if the category was previously deleted and added back with the same URL key.
- A customer can view a product from the Recently Viewed Products list if the product is associated with a different category than the one the customer is currently viewing.
- Any product page displays properly when the user accesses it from the site map on your web store.

This includes the case when an administrator sets the following option to **Yes: System > Configuration > CATALOG > Catalog > Search Engine Optimizations**, option named **Use Categories Path for Product URLs**.

- The category is not included in the URL to a product in your web store when an administrator sets following option to **No: System > Configuration > CATALOG > Catalog > Search Engine Optimizations**, option named **Use Categories Path for Product URLs**.

- The URLs for the same category and product, when assigned to multiple store views, are as expected.

Category URL example: */parent-category/category-2/*

Product URL example: */parent-category/category-2/product-1*

- With the option to Add Store Code to URLs option enabled, store codes properly display in URLs and the store view displays properly.

- After duplicating a product, the product's URL key updates successfully.

- A product URL key is correct in your web store even if there is a category with the same URL key as the product.

- Adding products without specifying a value for the **URL Key** field no longer results in the exception log entry **SQLSTATE[23000]: Integrity constraint violation**.

- You can create two or more categories with the same URL key without encountering the exception log entry **SQLSTATE[23000]: Integrity constraint violation**.

- The value of a product's **URL Key** field is properly validated.

A URL key can contain only alphanumeric characters (a-z, A-Z, 0-9) and the dash or hyphen character.

- A caching error that affected changing URL redirects was fixed.

- URL redirects work properly when products are assigned to categories in different store views but using the same request path.

- You can successfully import data and reindex with the import behavior set to Append Complex Data.

- After moving a category to another location in the category hierarchy, the category's URL and breadcrumbs update successfully.

- The options to create a custom redirect for a category and product work properly.

(In the Admin Panel, click **System > Configuration > CATALOG > Catalog > Search**

**Engine Optimizations.** In the right pane, from the **Create Permanent Redirect for URLs if URL Key Changed**, click **Yes**.

- Switching store views enables you to navigate to products and categories on that store view as expected. HTTP 404 (Not Found) errors no longer display and the specified category and product URL keys are correct.

In earlier EE versions, incorrect behavior was reported when the setting **System > Configuration > GENERAL > Web, Add Store Code toUrls** was set to **Yes**.

- A remote code execution vulnerability was fixed.
- On a clean installation, display errors are hidden.

## Magento EE 1.13.0.0 Release Notes

See the following sections for information about changes in this release:

- [Highlights](#)
- [Security Enhancements](#)
- [Security Advisory](#)
- [Upgrade Limitation](#)
- [Performance Improvements](#)
- [Tax Calculation Fixes](#)
- [API Fixes](#)
- [Fixes](#)
- [Changes](#)

## Highlights

- Major overhaul of tax calculation formulas, correction of rounding errors, and additional assistance with configuration
- Most indexing processes now run only to update products, categories, URL redirects, and so on that have changed—eliminating the need for manual full reindexing
- Additional option of using Redis NoSQL for cache and session storage in multi-host deployments (recommended for new deployments)  
To set up and use Redis with Magento, see [Using Redis with Magento Community Edition \(CE\) and Enterprise Edition \(EE\)](#).
- Full page caching now invalidates only pages that are affected by product or category changes
- Optimized cache adapters for single-server systems
- Elimination of many types of database deadlocks

## Security Enhancements

- The Magento Admin Panel and web stores no longer allow web browsers to store usernames or passwords.
- The Magento web store has additional Cross Site Request Forgery (CSRF) protections, meaning an imposter can no longer impersonate a newly registered customer and

perform actions on the customer's behalf.

- In earlier versions, Magento was vulnerable to a session fixation attack during the registration process. After logging in to their account, a registered user's session ID did not change. Therefore, if an attacker had knowledge of an unauthorized session ID and if that user successfully registers, the attacker was able to take over the newly registered account.  
Now, the session ID changes after successful registration, making unauthorized use of an account impossible.
- The cryptographic methods used to store passwords were improved to enhance security.

## Security Advisory

Magento has identified a potential vulnerability that might affect you if both of the following are true:

- Your store uses custom code that calls Magento full page caching functions
- You enabled the **Use SID on Frontend** configuration option  
(In the Admin Panel, **System > Configuration > WEB > Session Validation Settings, Use SID on Frontend.**)



**Note:** The potential vulnerability exists *only* if both of the preceding are true.  
Default Magento installations or installations that enable **Use SID on Frontend** but have no custom code with full page caching are at no risk.

If both of the preceding are true, Magento can be subjected to **cross-site scripting** (XSS) attacks—a type of injection issue, which means that malicious code is injected into otherwise trusted websites, generally in the form of a browser-side script.

**Issue:** Magento is subject to XSS attacks because the SID cookie value is not sanitized by default.

**Suggested solution:** Either disable **Use SID on Frontend** or **output-encode** any usage of the SID cookie value before using it or passing it as a parameter to any page cache helper functions.

Additional references:

- [XSS Prevention Cheat Sheet](#)
- [HttpOnly reference](#)
- [Open Web Application Security Project \(OWASP\) Code Review Guide V2.0](#)
- [Reviewing Code for Cross-Site Scripting](#)
- [HTTP Input and Output on php.net](#)

## Upgrade Limitation

Magento EE 1.13—unlike earlier versions—does not allow duplicate URL keys for products or categories. An issue has been identified that causes problems during upgrades if you already have duplicate URL keys. The issue is being addressed; until a solution is announced, Magento recommends you test your upgrade but do not try to deploy it to a production environment.

For more information about upgrading in a development environment, see [Upgrading to and Verifying Magento Community Edition \(CE\) 1.8 and Enterprise Edition \(EE\) 1.13](#).

## Performance Improvements

- Limited the way Magento performs large database lookups.
- Checkout performance improvements achieved by:
  -

Eliminating unnecessary calls to gift wrapping when loading the Shipping Method checkout step

- Eliminating unnecessary RSS cache cleanups when RSS functionality is disabled
- The locale used to send a new order confirmation e-mail now first checks to see if the customer's locale is the same as the store's locale before attempting to localize the e-mail.
- Improving the overall checkout process performance by loading the progress information for the current checkout step only
- You can load a large number of tax codes (35,000 or so) without impacting performance.
- Magento uses [MySQL database triggers](#) to improve access to the database during reindexing.

For more information, see [Magento Enterprise Edition 1.13 Benchmarking](#)

## Tax Calculation Fixes

Tax calculation issues can be divided into the following sections:

- [General Tax Notes](#)
- [Rounding Error Fixes](#)
- [Fixed Product Tax \(FPT\) Fixes](#)
- [Discount Calculation Fixes](#)
- [Display Fixes](#)

### General Tax Notes

The following general fixes were made to Magento tax configuration and calculations:

- Based on Magento testing and merchant experience, certain tax configuration settings have been determined to be susceptible to rounding issues and can be confusing to buyers. To help you avoid issues with those settings, warning messages display in the Admin Panel if you attempt to save such a configuration.  
Administrative users can choose to dismiss the messages and can still save the configuration; however, Magento strongly recommends you change the configuration in a way recommended by the details displayed in the message.  
For details, see the [Magento User Guide](#).
- Bundle pricing is more consistent as follows:
  - The calculation formula is: `Sub item price = Sub item base price * Applicable tiered price adjustment or discount, then rounded` `Bundle price = Sum (round(sub item price * qty))`
  - When non-integer quantities are multiplied by a product price, Magento rounds the resulting subtotal as follows: `round(unit price * non-integer quantity)`
- All product price information on which taxation is based are rounded to two digits of precision regardless of how many digits of precision have been loaded into the database (for example, \$10.24 instead of \$10.2385). This situation can occur when certain integrations enable third-party applications to send four-digit precision prices to Magento.  
Starting with this release those additional digits will have no impact on customer facing prices. Forcing two digits of precision enables more exact calculations involving Fixed Product Tax (FPT), discounts, and taxes—among other concerns.
-

For certain Canadian provinces and localities, calculations and methods were updated to support changing legal requirements in Canada:

- Provincial Sales Tax (PST)
- Goods and Services Tax (GST)
- Taxe de vente du Québec (TVQ)—also referred to as Quebec Sales Tax (QST)

For details, see the [Magento User Guide](#).

## Rounding Error Fixes

The following issues relate to one-cent rounding errors in the web store or shopping cart:

- Calculating taxes for bundled products with tiered pricing.
  - Calculating the price before customization for bundled products.
  - Calculating the grand total of items added to a cart in a different order.
  - Viewing an order when taxes are calculated after a discount using either row-based or unit price.
  - Applying a discount to an order with a shipping address different than the billing address.
  - Calculating the grand total based on the order in which products are added to the shopping cart.
  - Specifying that prices display in the web store excluding tax and setting a 20% tax rate (or discount rate) now calculates the grand total correctly. It is now possible to have grand totals in amounts like 6.99, 9.99, or 99.99—regardless of the currency units used in the web store.
  - Adding multiple items to a cart does not affect the accuracy with which taxation is calculated.
  - Subtotal (Incl. Tax) is now correct when catalog and shipping prices include tax. Both tax and discounts are applied after tax.
  - Prices displayed in the cart and on the catalog page are consistent and correct when catalog prices include tax, and when items in the catalog are set to display both including and excluding tax.
- (In the Admin Panel, click **System > Configuration > SALES > Tax**. In the right pane, expand **Calculation Settings**.)
- Error in calculating the Grand Total Excl. Tax was resolved. This error occurred in a specific configuration: tax is applied to FPT, FPT is included in the subtotal, and the customer selects non-taxable flat rate shipping.

## Fixed Product Tax (FPT) Fixes

The following issues relate to errors in calculating taxes that include FPT in the web store or shopping cart:

- Price in the cart displays the correct before-tax price and grand total.
- Subtotals displayed in the cart—both Including Tax and Excluding Tax—are now correctly calculated when FPT is applied.
- Free shipping offers are now processed correctly when FPT is applied.
- FPT taxes are calculated correctly when a discount is applied.

## Discount Calculation Fixes

The following issues relate to price calculations when coupon codes or other discounts are applied

in the web store or shopping cart:

- The Row Subtotal displayed in the cart is calculated correctly (that is, both Excl. Tax and Incl. Tax are correct).
- The price for bundled items now displays with tax included if the bundle is configured to do so.
- Taxation is now correctly calculated on a product with a discounted price.
- Taxation on discounts is now calculated correctly when the ship-to country is different from the web store's default country.

## Display Fixes

The following issues relate to the incorrect display of tax information in the Admin Panel or in your Magento web store:

- Row Subtotal displays correctly in the shopping cart when:
  - FPT is applied.
  - A discount is applied to a situation where the tax the customer pays is different from the tax specified for the web store's locale (for example, when the shipping origin is different than the shipping address).
- Subtotal including tax on a credit memo is correct when one or more items in the memo includes FPT.
- Item subtotal displays correctly when a discount is applied to a purchase that includes FPT.
- If the administrator sets catalog prices to exclude tax and to display product prices in catalog as including tax, the price of the product in your web store includes applicable taxes.  
(In the Admin Panel, click **System > Configuration > SALES > Tax**. In the right pane, expand **Calculation Settings**.)
- When Minimum Advertised Price (MAP) is enabled and the customer examines the price of a product in a gift registry, the price includes all of the following: price, actual price, price including tax, and price excluding tax.
- The amount of tax displayed in the Order Totals section of the shopping cart is now correct when free shipping and a shopping cart rule discount are applied.

## API Fixes

The following are fixed in the Magento SOAP v2.0 APIs (with exceptions noted):

- Requesting a product using a call like the following returns the product with the specified numeric SKU value (8888 in the following example):  
`$result = $client->call($sessionId, 'catalog_product.info', '8888', null, null, 'sku');`
- Order status is changed correctly using `salesOrderAddComment`.
- The `shoppingCartProductMoveToCustomerQuote` method works properly.
- You can now use `from-to` complex filters to perform "window" filtration on a single field. For example, you can use `from` and `to` on the `created_at` return a list of sales orders using the `salesOrderList`.
- When you use the SOAP API v.2.0 with `WS-I Compliance` enabled to retrieve sales orders information, the server responds with the correct `Content-Length` header.
- The `productGetSpecialPrice` method returns special price information for a product, whether or not `WS-I Compliance` is enabled.

- The `shoppingCartPaymentList` method returns the list of the available payment methods for the shopping cart appropriately. The following error is no longer returned: SOAP-ERROR: Encoding: object has no 'code' property in `name`
- The following issues with `WSDL` and `WS-I Compliance` are resolved:
  - The `productAttributeAddOption` and `catalogProductAttributeUpdate` methods are now supported when WS-I Compliance is enabled.
  - The WSDL declaration for `salesOrderCreditmemoCreate` is now correct; that is, it matches the code.
  - You can now add a C# web reference in Microsoft Visual Studio 2010 using the Magento WSDL.  
For example, this command no longer fails:  
`C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin> wsdl  
/out:Magento.cs /v http://magentohost/api/v2_soap/?wsdl`
  - The `WS-I Compliance` mode WSDL now includes `catalogProductAttributeInfo`.
- **XML-RPC API:** Using the `product_custom_option.add` operation with `multicall` no longer results in redundant options.

## Fixes

Fixes in this release can be divided into the following categories:

- [Web Store and Shopping Cart Fixes](#)
- [Promotional Price Rule Fixes](#)
- [Administrative Ordering and Credit Memo Fixes](#)
- [Import Fixes](#)
- [Payment Fixes](#)
- [Other Fixes](#)

### Web Store and Shopping Cart Fixes

- A customer's account created date is correct.
- When a product price is set with website scope and an administrative user has access to only one website, the default price is taken from that website scope. Also, when saving the product on the website scope, the price is updated only in that scope and not in the default scope.
- An error no longer displays on your web store after a customer places an order. (The error message was `There has been an error processing your request. Please contact us or try again later`).
- Restricted coupon codes work properly, even if the customer has selected the **Remember me** checkbox.
- Using the Table Rates shipping option, free shipping options work properly. (In the Admin Panel, click **System > Configuration > SALES > Shipping Methods**. In the right pane, expand **Table Rates**.)
- Issues with shipping table rates have been resolved.
- Reward Points are now granted one time per order, even when comments are added to the order later.

- Entering a value such as **10,50** (using a comma character and not a period) for **Adjustment Fee** now results in the correct amount of credit being applied to the transaction.
- Unit price for bundled products is now calculated correctly.
- The tiered price of bundled items now displays properly on the web store.
- Composite products can be successfully reordered.
- You can now use special characters in a product URL key.
- After a customer visits the sitemap, web stores URLs are no longer prepended by **/sitemap/catalog/string**.
- Welcome messages now display properly in the web store after a customer's profile information is changed.
- Recently viewed products now display updates properly.
- Armed Forces Middle East is now available for State when checking out.
- Gift wrapping charges now display properly in a PDF invoice.
- Searching for a customer's orders and returns works properly.
- Shipping is calculated correctly if you select **Using origin weight (few requests)** for **Packages Request Type**. (In the Admin Panel, click **System > Configuration > SALES > Shipping Methods > DHL (Deprecated)**.)
- Free shipping is no longer available to a customer during checkout if the option was disabled by an administrator. (In the Admin Panel, click **System > Configuration > Sales > Shipping Method > DHL(Deprecated)**, click one or more options from the **Allowed Methods** list, and, from the **Free Shipping with Minimum Order Amount** list, click **No.**)
- A user can navigate your web store while downloading a downloadable product.
- You can now specify weight units in kilograms (kg) using the FedEx shipping method.
- FedEx shipping rates are now consistent with Magento discounted rates.
- Fixed issues with United Parcel Service (UPS) shipping rates.
- An issue that caused a fatal error in the web store for an item in a gift registry has been resolved. The issue occurred when the item was removed from a website after a user had added the item to their gift registry.
- UPS shipping labels have the word **SAMPLE** printed on them only when you request a sample label.
- Changes made to United States Post Office (USPS) APIs and rates have been incorporated in Magento.
- You can now process a return materials authorization (RMA) for an order that was shipped to multiple addresses.
- The products in a customer's wish list no longer disappear after one or more products are edited by an administrator.
- Administrators can view the contents of a customer's shopping cart.
- The price of a simple product now displays properly when category permissions are enabled. (To set category permissions in the Admin Panel, click **Catalog > Categories > Manage Categories**, select a category, and click the **Category Permissions** tab.)

- Issues with purchasing a product using a gift card created with a custom option have been resolved.
- When a customer selects a product on your web store, the assigned category is selected in the navigation menu.
- With both flat index options enabled and set to update on save, mass attribute updates now display properly in your web store.  
For more information about flat catalog options, see the [Magento User Guide](#).
- With both flat index options disabled, after adding a product to many websites and assigning it to one or more categories, the product displays in the appropriate websites and categories in the web store.
- You can now use multiple selection attributes in a customer segment.

## Promotional Price Rule Fixes

The following fixes relate to administering and using shopping cart price rules and catalog price rules:

- Shopping cart price rules applied to specific customer groups work properly.
- Catalog price rules are applied properly to customer groups.
- The scope of a product attribute is now honored by a catalog price rule.
- Discounts specified by a shopping cart price rule are applied properly when a particular order is shipped to multiple addresses.
- You can add a gift card to an order that qualifies for a 100% purchase price discount specified by a shopping cart price rule.
- A discount specified by a shopping cart price rule that allows for more than one use per customer is applied the correct number of times if the customer has their orders shipped to more than one address.
- When an administrative user whose role is restricted to only viewing catalog price rules, the user cannot add or edit catalog price rules.
- Shopping cart price rules now work properly with bundled products.

## Administrative Ordering and Credit Memo Fixes

- When you create an order using the Admin Panel and you have multiple stores, the **State/Province** field updates appropriately for the country in which the order is placed.
- When you create an order using the Admin Panel and you have specified a default billing address and a default shipping address, the addresses are used correctly.
- Orders placed by an administrator display in a customer's last order list.
- Product comparisons now display properly when an administrator makes a change using the Admin Panel (for example, deleting a product from a customer's comparison list).
- You can now cancel an order using the Admin Panel.
- Orders and invoices that include taxable shipping—when created in the Admin Panel—now calculate the shipping taxes properly.
- Products added to a customer's wish list by an administrator display properly.
- Issues with the incorrect number of reward points being credited when issuing credit memos

have been resolved.

## Import Fixes

- The quantity (QTY) of all products imports correctly.
- The value of Maximum Qty Allowed in Shopping Cart (`use_cfg_max_sale_qty`) is correct.
- The product displays correctly in layered navigation.
- Importing customer lists with capitalization variations in the e-mail address now imports the customer only once (for example, `user@example.com` and `User@example.com`).
- Issues with importing products with **Append Complex Data** selecting from a comma-separated value (.csv) file have been resolved.

## Payment Fixes

- Resolved issue sending customer e-mail when using Payflow Link.
- Security issues with Google Checkout payments have been resolved.
- Security issues with Authorize.net payments have been resolved.
- Magento conforms to the latest version of the PayPal Instant Payment Notification (IPN) guidelines.
- The contents of a shopping cart are unaffected by canceling a PayPal payment.
- Issues with not being able to continue checkout after switching payment methods have been resolved.
- You can now process partial refunds and invoices for orders that were placed using Payflow Pro.
- Payflow Link and Payments Advance now capture IPN transactions properly.
- Special characters (such as e-mail addresses) are now handled properly by the Magento Payflow API integration.
- Resolved errors with orders placed using the Website Payments Pro payment method.
- PayPal Express Checkout payments are handled properly when a shopping cart price rule is specified.
- Any PayPal Name-Value Pair (NVP) payment method no longer automatically refunds an order when a chargeback is initiated. Magento now allows the dispute to be resolved before taking the appropriate action.  
PayPal NVP payment methods include: PayPal Payments Pro (including PayPal Payments Pro Hosted), Payments Standard, and all Payflow methods.
- PayPal Pro now correctly processes the shipping address for an order.
- PayPal Express Checkout and PayPal Pro now handle partial refunds properly.
- Fixed rounding errors that were preventing PayPal Express Checkout transactions from completing. The error occurred with the following configuration:
  - tax calculation method based on the total
  - tax calculated based on the shipping address
  - catalog prices exclude tax

- shipping prices exclude tax
- customer discount applied after a discount
- discount applied to prices excluding tax
- tax applied to a custom price if available

(In the Admin Panel, click **System > Configuration > SALES > Tax**. In the right pane, expand **Calculation Settings**.)

- The order status Suspected Fraud is now supported by PayPal Payments Pro (hosted) when PayPal fraud protection is enabled. Using the Magento Admin Panel, the merchant can also accept or deny any Suspected Fraud orders and have that decision applied to the PayPal transaction.
- When sending payments in the United Kingdom, PayPal Payments Pro (hosted) now sends the value for **state** correctly. (Before the fix, **city** was sent as the value for **state**.)
- Using the Ogone payment method, transactions display in the Magento Admin Panel after you capture them.
- When an administrator places an order and uses PSi Gate, then cancels the order, the PSi Gate gateway displays both the order and the void transactions.
- The following fields related to PayPal's Payflow Pro Gateway payment method are now implemented properly:
  - **[custref]** is the Magento customer's ID
  - **[INVNUM]** is Magento's order number
- Fixed spurious **Gateway error: Void error: V18A4B18E0F9** has been captured errors when canceling partially invoiced orders when the Payflow Pro processor was used to process the payment.
- **3-D secure fixes that affect UK merchants only:**
  - 3-D Secure for UK merchants implementing Direct Payment works properly.
  - SagePay Direct with 3-D secure payments are processed correctly.
- The Braintree payment method can now be configured properly.
- Partial captures are now supported for the following PayPal payment methods: Express Checkout, Payments Pro Payflow Edition, and PayPal Standard.
- Using the PayPal Express Checkout method, a recently added customer can check out without the error **This customer email already exists**.

## Other Fixes

- MySQL database deadlock issues were resolved.
- EE 1.13 is now [World Wide Web Consortium \(W3C\) compliant](#).
- A fatal error in **GiftRegistry/Model/Item/Option.php** has been resolved.
- When an administrative user whose role is restricted to managing products attempts to edit Inventory settings (**Catalog > Manage Products, Inventory**), only the available options display.
- Related product information updates appropriately in the Admin Panel.
-

Issues with editing product inventory settings and category attributes using the Google Chrome web browser have been resolved.

- Rolling back after a backup now works properly. (The Magento backup and rollback options are available in the Admin Panel in **System > Tools > Backup**.)
- You can now fetch data for a PayPal Settlement Report using a custom Secure FTP (SFTP) server.
- Using the Solr search engine with price navigation calculation step set to manual now displays search results properly on your web store. (A fatal error was fixed.)  
(To set price navigation step options in the Admin Panel, click **System > Configuration > CATALOG > Catalog > Layered Navigation**. From the **Price Navigation Step Calculation** list, click **Manual**.)  
For more information about using the Solr search engine with Magento EE, see the [Magento User Guide](#).
- You can now save a category with the option **Available Product Listing Sort By: Best value or Price** enabled.
- The following fixes relate to full page caching:
  - Breadcrumbs to a product work for all categories with which the product is associated.
  - The correct customer name displays on a gift card.
  - Product tags display properly.
  - Related products display properly.
  - *magento-install-dir/app/code/core/Enterprise/PageCache/Model/Config.php* was modified to enable you to set specific lifetimes for certain blocks.
- Automated e-mail marketing reminder rules work properly.
- Issues with session cookies have been resolved.

## Changes

This release includes the following changes to support the changes in indexing:

- As a result of the optimizations made to reindexing, you can no longer have a duplicate:
  - URL key for any two products
  - URL key for any two categories
  - Request Path for any URL Redirect (formerly referred to as *URL Rewrite*)
- There is a new management page in the Admin Panel: **System > Configuration > ADVANCED > Index Management**.
- The options on the **System > Index Management** are significantly different. In particular, because manual reindexing is no longer required for most indexers, there are fewer checkboxes and the page mostly displays status information.
- In the Admin Panel, **Catalog > Manage Products > edit-product > General** tab page option **Create Permanent Redirect for old URL if URL key changed** changes to **Create Custom Redirect for old URL**.

The feature behaves the same way; namely, selecting **Yes** creates a redirect for the old URL that points to the new URL if a page is moved.

**Note:** The wording of this option is expected to change back to the



previous wording in a subsequent release.

- The options **Product URL Suffix** and **Category URL Suffix** return an error if anything *except* alphanumerical characters or the underscore character are entered. (In the Admin Panel, click **System > Configuration > CATALOG > Catalog > Search Engine Optimizations**.)
- The `catalog_product_entity_url_key` and `catalog_category_entity_url_key` database tables for the corresponding `url_key` attributes have been added.

For more information about indexing changes, see the [Magento User Guide](#).

22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Installing and Upgrading to Magento Community Edition (CE) and Magento Enterprise Edition (EE)

## Table of Contents

- [Overview](#)
- [Getting Help With Your Installation or Upgrade](#)
- [Getting Magento Software](#)
- [Installing Magento CE or EE](#)
- [Upgrading to Magento CE or EE](#)

## Overview

We're happy you chose to install or upgrade your Magento software. We're going to help you to install or to upgrade to the following versions:

- Magento Community Edition (CE) 1.8 or 1.9
- Magento Enterprise Edition (EE) 1.13. or 1.14

These releases include a number of improvements:

- [Magento Community Edition \(CE\) Release Notes](#)
- [Magento Enterprise Edition \(EE\) Release Notes](#)
- [Magento CE and EE documentation home page](#), which also has information about new features

 **Important:** Use CE 1.9.1 or EE 1.14.1 or later for *all new EE installations and upgrades* to get the latest fixes, features, and security updates.

The sections that follow get you started on your installation or upgrade. Detailed, step-by-step procedures are discussed in the following articles:

- [Installing Magento CE or Magento EE](#)
- [Upgrading to Magento CE or EE](#)

## Getting Help With Your Installation or Upgrade

In the event you have a large, distributed system or you need additional help, consult the following resources.

| MAGENTO EDITION           | HOW TO GET HELP                                                                          |
|---------------------------|------------------------------------------------------------------------------------------|
| Magento CE and Magento EE | <ul style="list-style-type: none"><li>• <a href="#">Magento support forums</a></li></ul> |

- Twitter (@ask\_magento)

|                 |                                    |
|-----------------|------------------------------------|
| Magento CE      | <a href="#">Submit bug reports</a> |
| Magento EE only | <a href="#">Submit bug reports</a> |

## Getting Magento CE or EE

The following table discusses where to get the upgrade image and optional sample data.

| MAGENTO EDITION | UPGRADE IMAGE LOCATION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Magento CE      | <a href="http://www.magentocommerce.com/download">www.magentocommerce.com/download</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Magento EE      | <p>Use the following steps:</p> <ol style="list-style-type: none"> <li>1. Go to <a href="http://www.magento.com">www.magento.com</a></li> <li>2. In the top horizontal navigation bar, click <b>My Account</b>.</li> <li>3. Log in with your Magento username and password.</li> <li>4. In the left navigation bar, click <b>Downloads</b>.</li> <li>5. In the right pane, click <b>Magento Enterprise Edition &gt; Release</b> for the software or <b>Sample Data</b> for the optional sample data.</li> <li>6. Follow the instructions on your screen to complete the Magento EE download.</li> <li>7. Transfer the installation archive to your development system.</li> </ol> |

## Installing Magento CE or EE

See [Installing Magento CE or EE](#).

## Upgrading to Magento CE or EE

See [Upgrading to Magento CE or EE](#).

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Before You Install Magento: Recommended File System Ownership and Privileges

[Edit this page on GitHub](#) 

## Table of Contents

- [Overview](#)
- [Terminology](#)
- [Recommended Privileges and Ownership Before You Install Magento](#)
- [For More Information](#)

## Overview

In a continuing effort to improve security and ease of use, Magento is updating its recommendations for file system permissions and ownership for the following Magento editions:

- Magento Enterprise Edition (EE) versions 1.9 and later
- Magento Community Edition (CE) versions 1.4 and later

The guidelines discussed in this article apply to:

- New installations of or upgrades to the previously listed Magento versions only.  
The instructions might not work with older versions. If you have already installed Magento, see [After You Install Magento: Recommended File System Ownership and Privileges](#).
- Supported operating systems only.

 **Important:** This topic contains *suggestions* based on our experience. They are not *requirements* because we don't know the details of your deployment. Consult your hosting provider, a qualified security specialist, or an experienced system administrator for advice about specific security settings.

## Terminology

This article uses the following terminology:

### Hosted system

A Magento server located on a hosting provider. A *hosted* system typically does not enable you to elevate to `root`. The web server typically runs as an ordinary user. Magento assumes you log in as this user to start and stop the web server and that you already own all the files and directories in the Magento installation directory. You can use `chmod` to change permissions on files and directories.

### Dedicated system

A Magento server you control and operate. Unlike a hosted system, you can elevate to `root` and, as `root`, you can use the `chown` and

`chmod` commands to set ownership and privileges in the Magento installation directory.

## Recommended Privileges and Ownership Before You Install Magento

This section discusses Magento's pre-installation recommended privilege and ownership settings, which are as follows:

- The Magento installation directory and all subdirectories are owned by the web server user.  
This enables the web server to change files in these subdirectories but other users cannot access them (except a higher-level user such as `root`).
- All directories have 700 permissions (`drwx-----`).  
700 permissions give full control (that is, read/write/execute) to the owner and no permissions to anyone else.
- All files have 600 permissions (`-rw-----`).  
600 permissions mean the owner can read and write but other users have no permissions.
- If you use PHP-FPM (FastCGI Process Manager), make sure it is configured to run the same user as the Apache web server (Nginx web server).  
This allows web server to access content generated by Magento (Uploaded files, images, etc.).

**Note:** On a dedicated system, all commands discussed in this article must be entered as a user with `root` privileges.

After you extract the Magento installation package, set ownership and permissions as follows:

1. *Dedicated Magento server only.* Find the web server user:

- Apache:
  - Ubuntu: `grep User /etc/apache2/apache2.conf`
  - CentOS: `grep User /etc/httpd/conf/httpd.conf`

**Note:** The preceding paths are samples only. The paths to these `.conf` files on your system might be different. You can use the command `whereis nginx` to find the location of the configuration files.

Typically, the Apache web server user on CentOS is `apache` and the Apache web server user on Ubuntu is `www-data`.

- nginx: Open the nginx configuration file, typically `/etc/nginx/nginx.conf`. The `user` directive specifies the username. It might run as the Apache user if Apache is installed on the same system.

2. Change to the Magento installation directory.

3. *Dedicated Magento server only.* Enter the following command to set ownership of the Magento installation directory and all its subdirectories:

```
chown -R web-server-user-name .
```

For example, on Ubuntu where Apache usually runs as `www-data`, enter

```
chown -R www-data .
```

4. Enter the following commands to set directory permissions to 700 and file permissions to 600:

```
find . -type d -exec chmod 700 {} +
find . -type f -exec chmod 600 {} +
```

## For More Information

For more information about UNIX permissions, see the following resources:

- [UNIX File Permissions](#)
- [Unix Permissions Calculator](#)
- [Article on unix.stackexchange](#)

22 days ago

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# After You Install Magento: Recommended File System Ownership and Privileges

[Edit this page on GitHub](#) 

## Table of Contents

- [Overview](#)
- [Terminology](#)
- [Setting Privileges and Ownership After You Install Magento](#)
- [Securing Magento Extensions](#)
- [Applying Magento Support Patches](#)
- [For More Information](#)

## Overview

In a continuing effort to improve security and ease of use, Magento is updating its recommendations for file system permissions and ownership for the following Magento editions:

- Magento Enterprise Edition (EE) versions 1.9 and later
- Magento Community Edition (CE) versions 1.4 and later

This article discusses recommended permission and ownership schemes to apply after you install Magento.

The guidelines discussed in this article apply to:

- New installations of or upgrades to the previously listed Magento versions only.  
The instructions might not work with older versions. If you have already installed Magento, you can optionally change your file system permissions and ownership as discussed in this article.
- [Supported operating systems](#) only

This article discusses the following permission and ownership schemes:

- [Post-installation](#) for both a hosted Magento server and for a dedicated Magento server (in other words, you have `root` access on the server).
- [Securing extensions](#), recommended ownership and privilege settings to apply after you install Magento extensions.  
Typically, Magento extensions install with 777 (world-writable) privileges, which is undesirable from a security point of view.

 **Important:** This topic contains *suggestions* based on our experience. They are not *requirements* because we don't know the details of your deployment. Consult your hosting provider, a qualified security specialist, or an experienced system administrator for advice about specific security settings.

# Terminology

This article uses the following terminology:

## Hosted system

A Magento server located on a hosting provider. A *hosted* system typically does not enable you to elevate to `root`. The web server typically runs as an ordinary user. Magento assumes you log in as this user to start and stop the web server and that you already own all the files and directories in the Magento installation directory. You can use `chmod` to change permissions on files and directories.

## Dedicated system

A Magento server you control and operate. Unlike a hosted system, you can elevate to `root` and, as `root`, you can use the `chown` and `chmod` commands to set ownership and privileges in the Magento installation directory.

## Setting Privileges and Ownership After You Install Magento

If you have installed Magento, you can set file system privileges and ownership as follows:

- For a dedicated Magento server, you set ownership of files and directory as the web server user. You set privileges as 500 (directories) and 400 (files).
- For a hosted Magento server on which the web server runs as the logged-in username, you set privileges as 500 (directories) and 400 (files).



**Note:** In both hosted and dedicated systems, you set the privileges for the `media/` and `var/` directories at 700/600 because they must be writable.

Following is an explanation of the privileges:

- 500 permissions for directories (`dr-x-----`) gives the web server user read and execute privileges to prevent the accidental deletion or modification of files in the directory. Other users have no access to Magento directories.
- 400 permissions for files (`-r-----`) prevent any user (even the web server user) from overwriting files. This prevents attacks that depend on overwriting existing files with malicious content.
- 700 permissions (`drwx-----`) for the `media/` and `var/` directories give full control (that is, read/write/execute) to the owner and no permissions to anyone else.
- 600 permissions (`-rw-----`) for files in the `media/` and `var/` directories enable the web server user to write to them and to overwrite them.



**Note:** On a dedicated system, all commands discussed in this article must be entered as a user with `root` privileges.  
On a hosted system, commands must be entered as the web server user.

To set up ownership and permissions on a dedicated Magento server:

- Dedicated Magento server only.* As a user with `root` privileges, find the web server user:

- Apache:

- Ubuntu: `grep User /etc/apache2/apache2.conf`
- CentOS: `grep User /etc/httpd/conf/httpd.conf`



**Note:** The preceding paths are samples only. The paths to these `.conf` files on your system might be different. You can use the command `whereis nginx` to find the location of the configuration files.

Typically, the Apache web server user on CentOS is `apache` and the Apache web server user on Ubuntu is `www-data`.

- `nginx`: Open the `nginx` configuration file, typically `/etc/nginx/nginx.conf`. The `user` directive specifies the username. It might run as the Apache user if Apache is installed on the same system.
- Change to the Magento installation directory.  
On CentOS, this is typically `/var/www/html/magento`. On Ubuntu, it is typically `/var/www/magento`.
  - Dedicated Magento server only.* As a user with `root` privileges, enter the following command to set ownership of the Magento installation directory and all its subdirectories:

```
chown -R web-server-user-name .
```

For example, on Ubuntu where Apache usually runs as `www-data`, enter

```
chown -R www-data .
```

- Enter the following commands to set permissions:

```
find . -type f -exec chmod 400 {} +
```

```
find . -type d -exec chmod 500 {} +
find var/ -type f -exec chmod 600 {} +
find media/ -type f -exec chmod 600 {} +
find var/ -type d -exec chmod 700 {} +
find media/ -type d -exec chmod 700 {} +
chmod 700 includes
chmod 600 includes/config.php
```

## Securing Magento Extensions

If you set permissions and ownership as discussed in this article, you must change permissions temporarily to be able to use the Magento Connect Manager in the Admin Panel. (**System > Magento Connect > Magento Connect Manager**). You can still install extensions manually, however; that is beyond the scope of this article.

You can confirm the issue when you access Magento Connect Manager in the Admin Panel. The following error displays on the **Extensions** tab page:

```
Warning: Your Magento folder does not have sufficient write permissions.
```

To use Magento Connect Manager, you must:

1. Temporarily set 700/600 permissions on your Magento installation directory and subdirectories.
2. Install the extension.  
Magento Connect Manager typically installs extensions with 777 (world-writable) permissions.
3. Set permissions back to their recommended values.

In addition, if you have a dedicated Magento server, you should check ownership of files and directories and reset them if necessary. Often, Magento Connect Manager installs extensions with user and group ownership both set to the web server user.

## Temporarily Resetting Permissions on Your Magento Installation Directory

To temporarily set file and directory permissions so you can use Magento Connect Manager:

1. Change to the Magento installation directory.  
On CentOS, this is typically `/var/www/html/magento`. On Ubuntu, it is typically `/var/www/magento`.
2. Enter the following commands:

```
find . -type d -exec chmod 700 {} +
find . -type f -exec chmod 600 {} +
```

3. Install your extension using the Magento Connect Manager.

## Restoring the Recommended Permissions

Enter the commands discussed in this section to return permissions and ownership to their recommended values after you have installed extensions.

To restore Magento installation directory permissions:

1. Change to the Magento installation directory.  
On CentOS, this is typically `/var/www/html/magento`. On Ubuntu, it is typically `/var/www/magento`.
2. *Dedicated Magento server only.* As a user with **root** privileges, enter the following command to set ownership of the Magento installation directory and all its subdirectories:

```
chown -R web-server-user-name .
```

For example, on Ubuntu where Apache usually runs as `www-data`, enter

```
chown -R www-data .
```

3. Enter the following commands to set permissions:

```
find . -type f -exec chmod 400 {} +
find . -type d -exec chmod 500 {} +
find var/ -type f -exec chmod 600 {} +
find media/ -type f -exec chmod 600 {} +
find var/ -type d -exec chmod 700 {} +
find media/ -type d -exec chmod 700 {} +
```

## Applying Magento Support Patches

Magento Support typically provides a shell script to patch various Magento issues. When you run the shell script, file and directory permissions are typically not changed; however, the files provided with the patch are owned by the user who applied the patch. If you have

a dedicated Magento server, this is typically `root`; therefore, after applying the patch, you must change file ownership.

If you are required to apply a patch provided by Magento Support, use the following process:

1. Get the patch from Magento Support.

2. Follow the instructions provided with the patch.

Typically, you run a shell script as either a user with `root` privileges or as the owner of the Magento installation directory.

3. If you ran the patch as the owner of the Magento installation directory, you're done. File permissions aren't usually changed; however, you should check and reapply file and directory privileges if necessary.

4. If you ran the patch as a user with `root` privileges, use the following steps to reset file ownership:

5. *Dedicated Magento server only.* Find the web server user:

- Apache:

- Ubuntu: `grep User /etc/apache2/apache2.conf`

- CentOS: `grep User /etc/httpd/conf/httpd.conf`

Typically, the Apache web server user on CentOS is `apache` and the Apache web server user on Ubuntu is `www-data`.

- nginx: Open the nginx configuration file, typically `/etc/nginx/nginx.conf`. The `user` directive specifies the username. It might run as the Apache user if Apache is installed on the same system.

6. As a user with `root` privileges, enter the following command from the Magento installation directory:

`chown -R web-server-user-name .` For example, on Ubuntu where Apache usually runs as `www-data`, enter `chown -R www-data .`

## For More Information

For more information about UNIX permissions, see the following resources:

- [UNIX File Permissions](#)
- [Unix Permissions Calculator](#)
- [Article on unix.stackexchange](#)

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Installing Sample Data for Magento Community Edition (CE)

### Overview

This page discusses how to get the sample data for:

- Magento Community Edition (CE) 1.6, 1.7, or 1.8
- Magento Community Edition (CE) 1.9.0.0
- Magento Community Edition (CE) 1.9.1.0–1.9.2.0

 **Important:** Magento CE 1.9 and later use different sample data from the other versions in the preceding list. Make sure you use the correct version of sample data with your version of Magento CE.

### Install Sample Data Before You Install Magento

Magento CE sample data must be installed *before* you install Magento CE.

### Getting the Sample Data

The sample data is provided in different formats for your convenience. Archives for each version have exactly the same content (they differ only by compression method).

- [Magento CE 1.9.1.0–1.9.2.0](#)

### Installing the Sample Data

Detailed instructions for installing sample data for Magento CE 1.6–1.9 can be found on the [Magento Knowledge Base](#).

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Upgrading to and Verifying Magento Community Edition (CE) and Enterprise Edition (EE), Part 1

[Edit this page on GitHub](#) 

## Table of Contents

- [Overview](#)
- [Upgrade Path](#)
- [Understanding the Upgrade to EE 1.13.0.2 or Later](#)

## Overview

This article discusses how to upgrade to:

- Magento Community Edition (CE) 1.8 or 1.9
- Magento Enterprise Edition (EE) 1.13 or 1.14

## Upgrade Path

 **Important:** Magento recommends CE 1.9.1.0 or later or EE 1.14.1.0 or later for *all CE and EE installations and upgrades* to get the latest fixes, features, and security updates.

The following table provides basic information about how you perform your upgrade. More detailed information is discussed later in this article.

| EDITION AND VERSION | UPGRADE PATH                                    |
|---------------------|-------------------------------------------------|
| CE 1.4 or earlier   | Your current CE version > CE 1.7 > CE 1.9.0.0   |
| CE 1.5 or later     | Your current CE version > CE 1.9.0.0            |
| EE 1.7 or earlier   | Your current EE version > EE 1.12 > EE 1.14.0.0 |
| EE 1.8 or later     | Your current EE version > EE 1.14.0.0           |

## Magento EE Upgrade Path

Because of changes to URL rewrites, the upgrade to Magento EE 1.13.0.2 or later is more complex than other upgrades. (This includes upgrading to Magento EE 1.14.)

Follow are the specific versions affected:

- Upgrading from EE 1.12 or earlier to EE 1.13.0.2 or later (including to EE 1.14)
- Upgrading from EE 1.13.0.1 to EE 1.13.0.2 or later (including to EE 1.14)

For more information about these upgrades, see [Understanding the Upgrade to EE 1.13.0.2 or EE 1.14](#).



**Note:** If you're upgrading from EE 1.13.0.2 to EE 1.13.1.0 or later, your upgrade doesn't involve changes to URL rewrites and you can skip some of the steps discussed in the other upgrades. Continue with [Upgrading to and Verifying Magento Community Edition \(CE\) and Enterprise Edition \(EE\)—Part 2](#)

## Upgrading to Magento CE 1.9

Unless you're upgrading from an [older CE version](#), skip the remainder of this article and continue with [Upgrade Roadmap for Community Edition \(CE\)](#).

## Upgrading From Older Versions of Magento CE or EE

Provided you complete all tasks discussed in this article exactly as discussed, you can upgrade to Magento CE 1.9 or EE 1.14 from CE 1.5 or later or EE 1.8 or later.

**Important:** *Do not* upgrade directly to this release if you're currently running CE 1.4 or earlier or EE 1.7 or earlier. A safer approach would be:

1. Upgrade to CE 1.7 or EE 1.12
2. Thoroughly test CE 1.7 or EE 1.12 and fix any issues
3. Back up the system
4. Upgrade to CE 1.9 or EE 1.14



**Note:**

- Use caution when upgrading from versions older than CE 1.7 or EE 1.12 because core code on which your extensions are based and database schema have likely changed.
- To apply the [security patch \(October 2016\)](#) as part of your upgrade, see [How to Apply the SUPEE-8788 Patch](#).

To upgrade to CE 1.8, skip the remainder of this article and continue with [Upgrading to and Verifying Magento Community Edition 1.8 and Enterprise Edition 1.13—Part 2](#).

To upgrade from EE 1.13.0.2 to EE 1.13.1.0, skip the remainder of this article and continue with [Upgrading to and Verifying Magento Community Edition 1.8 and Enterprise Edition 1.13—Part 2](#).

## Understanding the Upgrade to EE 1.13.0.2 or EE 1.14

The following sections apply *only* to the following Magento EE upgrades:

- Upgrading from EE 1.12 or earlier to EE 1.13.0.2 or later
- Upgrading from EE 1.13.0.1 to EE 1.13.0.2 or later



**Note:** If you're upgrading from EE 1.13.0.2 to EE 1.13.1.0 or later, your upgrade doesn't involve changes to URL rewrites and you can skip some of the steps discussed in the other upgrades. Continue with [Upgrading to and Verifying Magento Community Edition \(CE\) and Enterprise Edition \(EE\)—Part 2](#)

## Why This Upgrade is Different

This section discusses why the upgrade to EE 1.13.0.2 or later requires you to perform different tasks than previous upgrades.

One of the results of the change to the way URL keys are handled is that URLs for the same products or categories might need to change from the URLs in the version you're upgrading from. Following are scenarios that might cause URL keys to change:

- All product keys must be unique. Any duplicate product URL keys are changed. For example, if two products had the URL key `nike-shoes`, one of them is changed to a URL key like `nike-shoes-1` by the URL redirect scripts. This also resolves any conflicts (for example, if you already had another product with key `nike-shoes-1`).
- Categories at the same level in the hierarchy in the same store view must have unique URL keys.  
For example, if you have a `shoes` category with URL key `shoes` in English store view, `schuhe` in the German store view and `schuhe` in the Swiss store view, the upgrade scripts change one of the category's URL keys to something like `schuhe-1`.

To ensure customers and search engines will be directed to the correct page, Magento provides a script that creates HTTP 301 (Moved Permanently) redirects for URLs that are changed.

In other words, after the upgrade, all pre-upgrade URLs work but they might be different than they were before you upgraded.

The way you upgrade depends on what EE version you're starting from. The following roadmaps provide a high-level overview of the upgrade process:

- [Upgrading From EE 1.12 or Earlier](#)
- [Upgrading From EE 1.13.0.1](#)

## Upgrading From EE 1.12 or Earlier

Following is a high-level roadmap to upgrade to EE 1.13.0.2 or later from EE 1.12 or earlier. This upgrade involves tasks not required for other Magento upgrades; these new tasks are indicated in the roadmap.

1. Install Magento in a different directory:
  - **Recommended.** Set up a new system (that is, another host) on which to install Magento. The system should be identical to, if not better than, your current system. The new system must meet the [Magento system requirements](#).
  - Install Magento in a new, empty root installation directory on the same server.



- In your current production environment:
    - a. Back up your Magento database.  
This includes the `media` directory and subdirectories; all extensions and customizations; and all custom themes.
    - b. Archive the file system.
  - In the development or test environment:
    - a. Create a new, empty database instance.
    - b. Import the production database tables into the development database instance.
    - c. Copy your production `media` directory, extensions, themes, and other customizations to the development system.
    - d. Copy `local.xml` to `[your Magento install dir]/app/etc` and edit it if necessary to reference the production database instance.
  - New** e. Stop all cron jobs.
  - f. In a web browser, go to your new base URL.
  - g. Wait for upgrade scripts to run.
  - New** h. Set all indexers to update when scheduled (**System > Configuration > ADVANCED > Index Management, Update when scheduled**).
  - New** i. Run the URL redirect script to create 301 redirects:

```
cd [your Magento install dir]
php -f shell/url_migration_to_1_13.php - thread-count
```
  - New** j. Run a full reindex from the command line:

```
[your Magento install dir]/php -f shell/indexer.php -- --reindexall
```
  - New** k. Clear the following directories from the command line:

```
[your Magento install dir]/var/cache /var/full_page_cache /var/locks
```
  - l. Enable cron and make sure the Magento cron job is set up.
  - m. Verify the upgraded system is now identical to the production system.  
If not, fix issues, retest, and upgrade again.
- *Test the upgrade thoroughly*, including:
    - Verify all extensions, themes, and customizations work.
    - Place orders using all webstores and all payment methods.

Step-by-step instructions for your upgrade start [here](#).

## Upgrading From EE 1.13.0.1

Following is a high-level roadmap to upgrade to EE 1.13.0.2 or later from EE 1.13.0.1. This upgrade involves tasks not required for other Magento upgrades; these new tasks are indicated in the roadmap.

1. Install Magento in a different directory:

- **Recommended.** Set up a new system (that is, another host) on which to install Magento. The system should be identical to, if not better than, your current system. The new system must meet the [Magento system requirements](#).
- Install Magento in a new, empty root installation directory on the same server.



**Important:** *Do not* upgrade Magento in the same directory on the same server because post-upgrade errors are likely to occur.

- In your current production environment:

- Back up your Magento database.

- Archive the file system.

This includes the `media` directory and subdirectories; all extensions and customizations; and all custom themes.

- In the development or test environment:

- Create a new, empty database instance.

- Import the production database tables into the development database instance.

- Copy your production `media` directory, extensions, themes, and other customizations to the development system.

- Copy `local.xml` to `[your Magento install dir]/app/etc` and edit it if necessary to reference the production database instance.

**New** e. Stop all cron jobs.

**New** f. Run the URL redirect script to create 301 redirects:

```
[your Magento install dir]/php -f shell/url_migration_from_1_13_0_0_to_1_13_0_2.php
```

- In a web browser, go to your new base URL.

- Wait for upgrade scripts to run.

**New** i. Run a full reindex from the command line:

```
[your Magento install dir]/php -f shell/indexer.php -- --reindexall
```

**New** j. Clear the following directories from the command line:

```
[your Magento install dir]/var/cache /var/full_page_cache /var/locks
```

- Enable cron and make sure the Magento cron job is set up.

- Verify the upgraded system is now identical to the production system.

If not, fix issues, retest, and upgrade again.

- *Test the upgrade thoroughly*, including:

- Verify all extensions, themes, and customizations work.

- Place orders using all webstores and all payment methods.

**Important:**

- Only after upgrade is tested and all errors fixed should you consider using it as your production system.
- Back up* your system frequently during the upgrade process so you can roll back to a previous state in the event of errors or issues.
- To avoid unnecessary downtime and potential issues, *never* upgrade Magento in a live environment!

Step-by-step instructions for your upgrade start [here](#).

22 days ago

• [Become a Contributor](#) • [Glossary](#) • [Privacy Policy](#) • [Terms of Service](#)

• [License/Trademark FAQ](#) • [Release Notes](#) • [Third-Party Licenses](#)

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento Open Source Release Notes (1.9 and later)

## Contents

These Release Notes contain the following information:

- [Important Upgrade Information](#)
- [Recent Patches](#)
- [Magento Open Source 1.9.4.5 Release Notes](#)
- [Magento Open Source 1.9.4.4 Release Notes](#)
- [Magento Open Source 1.9.4.3 Release Notes](#)
- [Magento Open Source 1.9.4.2 Release Notes](#)
- [Magento Open Source 1.9.4.1 Release Notes](#)
- [Magento Open Source 1.9.4.0 Release Notes](#)
- [Magento Open Source 1.9.3.10 Release Notes](#)
- [Magento Open Source 1.9.3.9 Release Notes](#)
- [Magento Open Source 1.9.3.8 Release Notes](#)
- [Magento Open Source 1.9.3.7 Release Notes](#)
- [Magento Open Source 1.9.3.6 Release Notes](#)
- [Magento Open Source 1.9.3.5 Release Notes](#)
- [Magento Open Source 1.9.3.4 Release Notes](#)
- [Magento Open Source 1.9.3.3 Release Notes](#)
- [Magento Open Source 1.9.3.2 Release Notes](#)
- [Magento Open Source 1.9.3.1 Release Notes](#)
- [Magento Open Source 1.9.3.0 Release Notes](#)
-

- [Magento Open Source 1.9.2.4 Release Notes](#)
- [Magento Open Source 1.9.2.3 Release Notes](#)
- [Magento Open Source 1.9.2.2 Release Notes](#)
- [Magento Open Source 1.9.2.1 Release Notes](#)
- [Magento Open Source 1.9.2.0 Release Notes](#)
- [Magento Open Source 1.9.1.0 Release Notes](#)
- [Magento Open Source 1.9.0.1 Release Notes](#)
- [Magento Open Source 1.9.0.0 Release Notes](#)
- [Open Source Software Licensing Agreements](#)

## Important Upgrade Information



**Important:** Use Magento Open Source 1.9.3.0 or later for *all new Magento Open Source installations and upgrades* to get the latest fixes, features, and security updates.

## Magento Open Source 1.9.4.5 Release Notes

This version (or patch SUPEE-11314, which applies to older versions of Magento) provides resolution of multiple critical security issues. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

We recommend upgrading your Magento store to this latest version. See [Magento | APSB20-22](#) for a comprehensive discussion of these issues.

## Magento Open Source 1.9.4.4 Release Notes

This version (or patch SUPEE-11295, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

We recommend upgrading your Magento store to this latest version. See [Magento | APSB20-02](#) for a comprehensive discussion of these issues.

## Fixed issues and enhancements

- The **Disable** button present when you run the compiler from **Admin > System > Tools > Compiler** is now enabled as expected. Previously, when you clicked the **Disable** button, it did not change state.

## Magento Open Source 1.9.4.3 Release Notes

This version (or patch SUPEE-11219, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

## Fixed issues and enhancements

- WebserviceX has been removed from the Magento 1.x code base.
- This release adds two new currency services for currency rate import: [CurrencyConverterAPI](#) and [FixerIO](#)

## Known issue

This release includes a fix for a security vulnerability that potentially allowed changes to protected store settings. As a result, extensions or customizations that depend on saving configuration fields that are not defined in `system.xml` files may no longer work correctly.

## Magento Open Source 1.9.4.2 Release Notes

This version (or patch SUPEE-11155, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

**Note:** We are aware of the incompatibilities between patch SUPEE-11155 and the PHP 7.2 support patch and are currently working on a new version of SUPEE-11155 that resolves these incompatibilities. See [Security Patch SUPEE-11155 - Possible issues?](#) for a community-driven discussion on issues and solutions related to this SUPEE. Check these release notes and the [Magento Security Center](#) for updates on the availability on the new patch.

We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

## Fixed issues and enhancements

- The Magento logging feature now works as expected after the SUPEE-11086 patch is installed. Previously, after application of this patch, Magento could only write only to a file that already existed on the server, and did not create new log files.
- Magento 1.14.4.0 and the PHP7.2 support patch now include the same files as expected. The previous version of the patch did not include the following three files, which were included in Magento 1.14.4.0. Magento 1.14.4.0:  
`lib/phpseclib/PHP/Compat/Function/array_fill.php`,  
`lib/phpseclib/PHP/Compat/Function/bcpowmod.php`, and  
`lib/phpseclib/PHP/Compat/Function/str_split.php`.

## Known issues

The extensive security enhancements we've included to this release have resulted in the following changes to Magento behavior:

- You can no longer upload files with the extension `.swf` to the WYSIWYG editor.
- Third-party checkout extensions and closed security cases will either not work securely or will not work at all.
- The Authorize.net Direct Post module has been enhanced to support the replacement of Authorize.net's MD5-based hash with a (SHA-512) signature key. Authorize.net will no longer support implementations using the MD5-based hash as of June 28, 2019. You will need to update your signature key after upgrading to this version of Magento. For information about updating your signature key, see the Get a New Signature Key discussion in the [Update Authorize.Net Direct Post from MD5 to SHA-512](#) help article. Note that although this help article describes how to install the earlier patch, merchants upgrading to this release of Magento are not applying the patch and should consult only the Get a New Signature Key discussion. If you've applied the patch to your Magento installation while running an earlier version of

Magento, uninstall the Update Authorize.Net Direct Post from MD5 to SHA-512 patch before upgrading to this release.

- You can no longer preview JavaScript in a newsletter template in the Admin.
- Sitemap names cannot exceed 32 characters.

## Magento Open Source 1.9.4.1 Release Notes

This version (or patch SUPEE-11086, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

**Note:** Magento's implementation of the Authorize.Net Direct Post payment method currently uses MD5-based hash for all M1 and M2 installations. As of June 28, 2019, Authorize.Net will stop supporting MD5-based hash usage.

This will result in Magento merchants not being able to use Authorize.Net Direct Post to process payments. To avoid disruption and to continue processing payments, merchants must apply a patch provided by Magento and add a Signature Key (SHA-512) in the Magento Admin configuration settings. Magento released this patch in late February to address this issue on pre-2.3.1 installations of Magento. See [Update Authorize.Net Direct Post from MD5 to SHA-512](#).

Information about the deprecation of Authorize.Net Direct Post can be found in [Authorize.net Direct Post \(Deprecated\)](#).

## Fixed issues and enhancements

- Google Image Charts has been deprecated and replaced by [Image-Charts](#) for dashboard charts.
- Layered navigation now works as expected when full page cache and block caching are enabled. Previously, you could not clear layered navigation filters when these features were enabled.
- Errors caused by problematic PHP error logging have been resolved. Previously, Magento displayed excessive and unnecessary 404 errors.
- Magento now displays the following message when an invalid character is used, **Attribute code is invalid. Please use only letters (a-z), numbers (0-9) or underscore(\_) in this field, first character should be a letter. Do not use "event" for an attribute code.** Previously, Magento did not flag invalid attribute codes.
- You can now add to the cart products with custom options for which the **custom option** checkbox has not been checked. Previously, Magento did not add the product to the cart, and displayed this message, **Cannot add the item to shopping cart**.
- URL redirects for products now work as expected. Previously, when you selected a product from the Category page and Add URL Redirect has been enabled, Magento redirected users to URL Redirect Information and threw this error, **exception 'Mage\_Core\_Exception' with message 'Invalid block type: Mage\_Adminhtml\_Block\_Empty\_Edit\_Form'** in app/Mage.php:580
- Magento now displays payment information during the confirmation step of check out and successfully processes an order when inline translation is enabled. Previously, Magento did not display this payment information during check out, and the order was not completed.

- You can now create a staging website when development mode is enabled. Previously, Magento threw an error after you added a website from System > Content Staging > Staging Websites.
- You can now successfully delete a website by clicking **Delete Website** as expected. Previously, when you clicked this button, Magento threw a fatal error.
- You can now add a banner by clicking **Add Banner** from the Admin. Previously, Magento threw an error when you clicked this button.
- Magento no longer throws an `Undefined index: is_recurring` error when you try to save a product when deploying Magento with development mode enabled.

## Known issue

- If there were any custom widgets created by usage of helper methods in layout updates, they will no longer work.

# Magento Open Source 1.9.4.0 Release Notes

This version (or patch SUPEE-10975, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These critical security issues include remote code execution (RCE), cross-site scripting (XSS), and cross-site request forgery (CSRF) issues. This release also provides support for PHP 7.2.

We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

Note that after updating to this release, third-party modules that depend upon Magento core backup functionality will no longer work. Alternatively, you can use one of these two methods to enable database backups:

- Change `false` to `true` in the **config > modules > Mage\_Backup > active** setting of `app/etc/modules/Mage_All.xml`.
- From the Admin, change the **System > Configuration > Advanced > Disable Modules Output > Mage\_Backup** from `disabled` to `enabled`.

## Fixed issues and enhancements

- This release provides support for PHP 7.2.
- We've removed the CC module. As a result, third-party modules that depend upon either the `ccsSave` method or the `xmlConnect` method will not work as expected. Third-party themes that implement `ccsSave` will not work as expected, either.
- The Magento logo has been updated throughout the code base.
- The **Continue** button now works as expected on the Payments step of checkout when paying with the PayPal payment method.
- Google Tag Manager now logs sales information in Google Analytics as expected.
- The product export CSV file now contains columns for super attributes.
- Magento no longer throws an error when a customer accesses their shopping cart after items in their cart have been removed due to a timeout. Previously, Magento displayed this error, `'Notice: Undefined variable: freePackageValue in /var/www/dev/htdocs/app/code/core/Mage/Shipping/Model/Carrier/Tablerate.php on line 130'`.
- Clicking on a configurable product's swatch on the product list page now updates product

price as expected.

- Customers can now successfully add a grouped product to their shopping cart when category permissions are enabled. Previously, Magento did not add the product to the cart, but instead displayed a descriptive error message.
- Magento no longer displays incorrect prices on the storefront after a failure of the enterprise refresh index.
- We've resolved issues in the indexing locking mechanism that previously resulted in Magento throwing an exception after indexing completed.
- Magento no longer throws a fatal error when a merchant uses an already reserved word to name a product attribute.
- Magento now adds the correct sales tax to orders being shipped to U.S. addresses that use zip codes with the optional four-digit suffix (for example, 73365-1234). Previously, the Tax rule triggered a failure if the U.S. zip code that had this optional four-digit suffix.
- Magento now displays all products on a production website that were edited by a role-restricted user on the associated staging website.
- We've resolved an issue that caused Target Rules to throw an exception when a customer opened a product view page.

## Magento Open Source 1.9.3.10 Release Notes

This version (or patch SUPEE-10888, which applies to older versions of Magento) provides resolution of multiple critical security issues. These critical security issues include remote cross-site scripting and cross-site request forgery issues. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

**Note:** With this release, Magento is announcing the following support policy: For Magento Open Source 1.5 to 1.9, Magento will provide software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

### Known issue

You cannot re-send the password for new customers who created their account during checkout.

## Magento Open Source 1.9.3.9 Release Notes

This version (or patch SUPEE-10752, which applies to older versions of Magento) provides resolution of multiple critical security issues. These critical security issues include remote code execution, cross-site scripting, and cross-site request forgery issues. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

### Fixed issues and enhancements

- Magento no longer performs unnecessary write operations on the `core_url_rewrite` table.
- Customers can now successfully register during checkout without being unexpectedly logged out.
- Incorrect escaping in the `cron.sh` file no longer prevents cron jobs from running in parallel as expected.
- Magento now cleans session data as expected after a customer logs out.

## Known issue

If your custom code or extension is using `Zend/Filter/PregReplace.php` with the modifier `e`, it will now return an error due to possible RCE issues. See [Magento Security Center](#) for more information.

# Magento Open Source 1.9.3.8 Release Notes

This version (or patch SUPEE-10570, which applies to older versions of Magento) provides resolution of multiple critical security issues. These critical security issues include authenticated Admin user remote code execution, unauthorized data leaks, and cross-site request forgery (CSRF) vulnerabilities. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

## Fixed issues and enhancements

- Changed Magento Admin to support recent USPS shipping changes. On February 23, 2018, USPS removed APIs that support the creation of shipping labels without postage. In response, we've removed this functionality from the Magento Admin. Consequently, you cannot create and print shipping labels that do not have postage applied.
- Updated copyright to 2018.

## Known issues

These two known issues are associated with the use of HTML tags within a product's SKU attribute:

- If you try to import products that contain HTML tags in the SKU attribute, Magento displays this error at the data validation stage (that is, when you click **Check data**): **Invalid value in SKU column. HTML tags are not allowed.**
- If you try to create or edit a product in the Admin panel and the product's SKU attribute value contains HTML tags, Magento throws this error when you try to save the product: **HTML tags are not allowed in SKU attribute.**

# Magento Open Source 1.9.3.7 Release Notes

This patch (SUPEE-10415) provides resolution of multiple critical security issues. These critical security issues include remote code execution, cross-site scripting, and cross-site request forgery issues. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

## Fixed issues

- Magento no longer displays the "Invalid Secret Key. Please refresh the page." message when a user loads the Admin.
- The one-page checkout page now displays the following message when a customer checks out an order for which no amount is due: **No payment information required.** Magento versions prior to 1.14.3.3 included this message, but it was missing from v1.14.3.3.
- We've fixed a typo in the patch header information. (`autocomplete="new-pawsord"` is now `autocomplete="new-password"`.)

## Known issue

**Issue:** Magento displays a "404: Page Not Found" error from the `errors/` directory after upgrading to SUPEE-10415. This issue occurs only in Magento installations that run certain third-party

extensions.

**Description:** Magento is not properly logging PHP warnings that occur early during page initialization. Instead, of logging the error and continuing operation, Magento generates a 404 page. (Previously, Magento logged these warnings in the system.log file, and execution would continue as usual.)

**Workaround:** Confirm that there are no PHP warnings generated by any of the extensions or customizations.

## Notes

- We no longer support custom file extensions for `Mage::log()`. Supported file extensions include `.log`, `.txt`, `.html`, `.csv`. For more information, navigate to **Developers > Log Settings** from the Admin. Magento displays this comment: **Logging from Mage::log(). File is located in /var/log. Allowed file extensions: log, txt, html, csv.**
- Passwords for new users are now limited to 256 characters. If a new user enters a password that exceeds 256 characters, Magento displays this message: **Please enter a password with at most 256 characters.**

## Magento Open Source 1.9.3.6 Release Notes

This patch (SUPEE-10266) provides resolution of multiple critical security issues and several functional fixes. These critical security issues include remote code execution, cross-site scripting, and cross-site request forgery issues. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

- We've also fixed an issue where uploaded images were twice their original size after you applied SUPEE-9767 v2.

## Magento Open Source 1.9.3.5 Release Notes

We have skipped release 1.9.3.5.

## Magento Open Source 1.9.3.4 Release Notes

This patch addresses both security and functional issues discovered when using the **SUPEE-9767** patch. We recommend upgrading your Magento store to this latest version.

Here are your upgrade options:

- Upgrade to Magento Open Source 1.9.3.4. You don't need to revert any patches or install SUPEE-9767 version 2 as version 2 is already included in the Open Source 1.9.3.4 release.
- If you've already installed SUPEE-9767 version 1, you can either:
  - Upgrade to Magento Open Source 1.9.3.4, or
  - revert SUPEE-9767 version 1, then install SUPEE-9767 version 2.
- If you haven't installed SUPEE-9767 version 1, you can either:
  - Upgrade to Magento Open Source 1.9.3.4, or
  - install SUPEE-9767 version 2.

See [Magento Security Center](#) for a comprehensive discussion of these security issues.

This release also provides support for the following functional issues:

## General fixes

- We've restored missing `strip_tags` functionality in the checkout JavaScript.
- We've changed how Magento validates form keys during the generic five-step checkout process. Previously, customer registration failed during standard checkout processing if form key authentication was enabled.
- Magento now displays the `Allow_symlinks` message in the Admin message area as expected.
- Magento now preserves the background transparency of uploaded images as expected. Previously, transparency was lost after the image was uploaded, resulting in an unusable image.
- You can now use Checkout with Multiple Addresses when checkout form validation is enabled.

## Installation

- The Allow symlinks option is now disabled during installation or upgrade processes. Previously, when you changed the Allow symlinks setting to true in the database before upgrading and then installing the patch, this option remained enabled, but you could no longer access it from the Admin panel.

This patch is available from [Magento Tech Resources](#).

## Magento Open Source 1.9.3.3 Release Notes

This patch provides resolution of multiple critical security issues. These critical issues include remote code execution for authenticated Admin users, access control bypass, and cross-site request forgery issues. See [Magento Security Center](#) for a comprehensive discussion of these issues.

This release also provides support for the following issue:

Support for PayPal's update to its Instant Payment Notification (IPN) server URL. PayPal provides more information about this feature in [IPN Verification Postback to HTTPS Microsite](#). This update is essential for retaining uninterrupted service after June 30.

SUPEE-8167, an older patch that also contains this fix, was added on May 8, 2017, and is available from [Magento Tech Resources](#).

## Known Issues

This patch/release has known issues. Please see [SUPEE-9767](#) for updates.

**Note:** Before applying this patch or updating to this release, disable the **Symlinks** setting in **System > Configuration > Advanced > Developer > Enable Symlinks**. If the **Symlinks** setting is enabled, it will override your configuration file settings. If that override occurs, you will need to directly modify the database to change those settings.

## Magento Open Source 1.9.3.2 Release Notes

This patch addresses the following issues:

- Removal of vulnerability with the Zend framework `Zend_Mail` library. For more information, see [Magento Security Center](#).
- Updated the copyright year to 2017.

# Magento Open Source 1.9.3.1 Release Notes

This patch addresses the following issues:

- We restored the old tax calculation algorithm for shipping charges. The patch to apply new calculation will be available on request.
- Resolved an issue with setting the session lifetime to 0.
- The monthly cron job that cleans up the table that contains IP addresses and passwords runs properly.
- All configurable product images are imported.
- You no longer get an exception due to an undefined `addCrumb()` method call.
- Resolved the error `Notice: Undefined index: session_expire_timestamp` when accessing the storefront.
- Values for drop-down label values are saved correctly.
- The "Price as configured" for bundle products displays correctly in the shopping cart.
- Auto-generated passwords are sent to new customers as expected.
- The method `Mage_Api_Model_Server_Handler_Abstract::processingMethodResult()` accepts scalar and array values.
- The default MySQL Full-Text search works as expected; it no longer returns all products.
- Prevented a potential [Cross-Site Request Forgery \(CSRF\)](#) vulnerability by changing the form key when a customer signs out of the storefront.
- Catalog price rules return the correct price.
- Indexers now update all products instead of skipping the last product updated.



**Note:** You currently *cannot* upgrade to this version using Magento Connect Manager. We expect to resolve this issue soon.

# Magento Open Source 1.9.3.0 Release Notes

See the following sections for information about this release:

- [Highlights](#)
- [Security Enhancements](#)
- [Check for .swf Files After Upgrade](#)
- [Backward-Incompatible Changes](#)
- [Fixes](#)

## Highlights

Magento Open Source 1.9.3 delivers more than 120 quality improvements, as well as support for PHP 5.6 in addition to PHP 5.4 and 5.5.

## Security Enhancements

We addressed the following security issues in this release:

- General security enhancements
- Password enhancements

## General security enhancements

For more information about these security enhancements, see [our Security Center article](#).

Patches for major security issues in earlier versions of the Magento software are available on the [Magento download page](#) (look for **SUPEE-8788**.)

See [How to Apply the SUPEE-8788 Patch](#)

- Resolved a potential SQL injection (Zend Framework issue)
- Resolved a cache poisoning issue
- We now provide better protection against path exploits.
- Resolved a potential [cross-site scripting \(XSS\)](#) vulnerability when adding a category.
- Resolved a potential XSS vulnerability that affected the Magento server's request URI.
- Resolved a potential XSS vulnerability in invitations.
- You can no longer cause out-of-memory errors on the Magento server by flooding it with images that have incorrect dimensions.
- The Magento Admin Panel login page now renders in HTTPS if you configured the Magento server for HTTPS.
- We added the [nosniff header](#) to our .htaccess files.
- Magento no longer uses Adobe Flash for uploads.
- Fixed several potential issues indicated by static code scans.
- Resolved a potential man-in-the-middle vulnerability.
- Resolved a potential PHP security vulnerability.
- An administrative user is no longer able to create a potential security vulnerability that used the block cache.
- Resolved a potential [cross-site request forgery \(CSRF\)](#) vulnerability involving the wishlist.
- Resolved a potential remote code execution exploit.
- It is no longer possible to log in to a store as an existing customer using only an e-mail address.

## Password enhancements

- A user can reset a password only after receiving an e-mail. In addition, we introduced the following configuration settings:
  - Limit the number of forgotten password requests from one IP address to five times per hour.
  - Limit the number of forgotten password requests from one e-mail address to five times per 24 hours.
  - Limit the number of forgotten password requests to no more than once ever 10

minutes per e-mail address.

- The forgot password link expires after the first use or two hours (by default).
- When a user changes their e-mail address, they are required to provide their password and to acknowledge the change from the previous address.
- We now ignore leading and trailing spaces in a user's password.
- The new customer e-mail now includes the customer's password.
- Resetting a password using a password recovery e-mail succeeds.

## Check for .swf Files After Upgrade

If you upgraded to Magento Open Source 1.9.3 after applying the [SUPEE-8788 patch](#), make sure the following files have been deleted:

```
skin/adminhtml/default/default/media/flex.swf
skin/adminhtml/default/default/media/uploader.swf
skin/adminhtml/default/default/media/uploaderSingle.swf
```

If the files are present, delete them to avoid a potential security exploit. As of Magento Open Source 1.9.0.0, we no longer distribute .swf files with the Magento software.

## Backward-Incompatible Changes

The following backward-incompatible changes were made in this release:

`Mage_Adminhtml_Block_Cms_Wysiwyg_Images_Content_Uploader`: Parent class was removed.

`Mage_Uploader_Model_Config_Abstract`: Overrides the magic method `__call` and its behavior can be inconsistent. For example:

```
->setData('underscore_key', 1)
->getUnderscoreKey() //null
```

## Fixes

The following sections discuss other fixes in this release:

- [Tax Calculation Fixes](#)
- [Shopping cart and checkout fixes](#)
- [Catalog fixes](#)
- [Price rule fixes](#)
- [Configurable swatches fixes](#)
- [Import/export fixes](#)
- [Indexer fixes](#)
- [Other fixes](#)

## Tax Calculation Fixes

- The subtotal including tax on an invoice is calculated correctly.

## Shopping cart and checkout fixes

- One product displays one time in a cart even if the product was added once as a guest and another time as a logged-in user.
- Bundled products now display properly in the mini cart as well as the shopping cart.
- Moving a configurable product to a shopping cart in the Admin Panel functions normally.
- Shipping discount coupons are now based correctly on a customer's shipping address.
- First Class Mail letter now displays as a shipping option in the shopping cart.
- You can now pay for a product using both store credit and reward points.
- An exception no longer displays when a customer uses a gift card in an invalid transaction (such as an incorrect payment card number).
- We added validation so a special price must be less than the actual price.
- Exceptions no longer display when a customer checks out.
- Fixed a programming issue that prevented serializing and unserializing values in the shopping cart.
- Magento recovers from payment processor unavailability properly; the customer is charged and the item is shipped.
- You can no longer order an empty product; that is, a product with no options.

## Catalog fixes

- A configurable product with decimal quantity less 1 now displays the proper quantity in the catalog.
- Configurable products are now sorted by attribute, not by product ID.
- Errors no longer display when you use `Mage_Catalog_Block_Product_List` on a product detail page.
- Removed the undefined variable `where` from  
`app/code/core/Mage/CatalogSearch/Model/Resource/Fulltext.php`.

## Price rule fixes

- A catalog price rule that targets a bundled product by percentage calculates the price properly.
- A shopping cart price rule that includes tax now calculates properly.
- With the flat product catalog enabled, a catalog price rule with multi-select attributes works properly.
- Errors no longer display when two users add a product at the same time. Magento thanks Babenko eCommerce for contributing this fix.
- You can now add configurable products to the shopping cart after configuring a shopping cart rule.

## Configurable swatches fixes

- Fixed a memory leak in the configurable swatches module.

- Configurable swatches for out-of-stock products now display consistently in layered navigation, the category view page, and the product view page.
- Configurable swatches work properly even if there is no image (before the fix, a JavaScript error was thrown).
- Resolved performance issues.
- Swatch images for configurable products display properly.

## Import/export fixes

- We bundled the following fixes in a patch:
  - Exporting a large number of products no longer results in an out-of-memory error.
  - You can import into multiple stores if some stores are set to be replaced.
  - Re-importing customers that have a multi-select attribute preserves the attribute.
  - File uploads are processed properly.
  - Fixed broken help links in the Magento Admin Panel.
- Importing products no longer consumes an excessive amount of memory.
- Coupon reports exported as .csv now display the correct totals.

## Indexer fixes

- With flat category tables enabled, reindexing no longer removes the category class tag.
- Resolved errors with the Product Flat Index not completely indexing a large number of changes.
- All indexes now reindex when set to update when scheduled.
- Improved performance of the category indexers. Magento thanks Vaimo for contributing this fix.
- Categories saved with a / character as the suffix display properly.

## Other fixes

- Applied United States Postal Service API changes for [January 17, 2016](#).
- Default variable values now save normally.
- The WYSIWYG editor handles XHTML tags like cellpadding and cellspacing properly.
- The configuration setting **Allow HTML Tags on Frontend** is honored.
- Orders created using the Magento Admin now display on the Orders and Returns page on the storefront.
- The option to merge Cascading Stylesheets (CSS) and JavaScript now works properly with a responsive web design (RWD) theme.
- On a mobile device when the Magento storefront uses an RWD theme, the Filter bar displays one time only.
- The Magento Connect Manager downloader's .htaccess file is no longer overwritten when the

downloader component is updated.

- The configuration cache is no longer corrupted under heavy load.
- Order update e-mails are sent only once.
- A SOAP API call to `/api/soap/?wsdl` returns normally.
- A value that contains special characters is handled without errors by the SOAP API.
- Fixed the untranslatable `base/default/template/sales/guest/form.phtml` template.
- Magento now stores two-digit birth years properly (for example, `80` is stored as `1980`).
- HTTP 200 (OK) status codes are returned for pages after a session expires.
- You can view a disabled product without errors if compilation is enabled.
- A Value Added Tax (VAT) ID now validates properly. If the customer specifies an invalid ID, the customer is notified they will be charged VAT tax.
- Listing shipments no longer displays an exception.
- You can filter associated products for a group product without errors.
- When you manage product attributes, selecting an action from **Actions** works properly.
- You can now add a configurable product by SKU to an order using the Admin Panel.
- You can now save a product's weight attribute.
- Using a Portable Network Graphics (`.png`) image on a CMS page no longer results in a `HEADERS_ALREADY_SENT` message to be logged.
- Fixed an exception related to an unknown database table.
- Fixed an issue with JavaScript merging.
- You can now print 10 or more shipping labels without issues.
- A PHP notice no longer occurs when you log the Magento Admin Panel IP address in the event log.
- A SQL error no longer displays when you create a new floating point product attribute programmatically.
- Added a missing image to the codebase.
- The expression `Mage::getModel('core/variable')->addValuesToResult()` returning a collection with column `plain_value` and `html_value` now returns a collection with columns `plain_value` and `html_value`.
- Payment no longer results in the exception `ERR (3): Notice: Undefined offset: 1 in app/code/core/Mage/Sales/Model/Order.php on line 1258`.
- The correct telephone number displays in transactional e-mails. We changed the variable `phone` to `store_phone`. Magento thanks Classy Llama Studios for contributing this fix.
- The Google sitemap now lists store URLs properly.
- Implemented search query caching, which speeds up search results.
- After a customer submits an order, the following error should not display: `SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'EC00000148' for key`

'UNQ\_SALES\_FLAT\_ORDER\_INCREMENT\_ID'.

- With flat category enabled, you no longer see errors due to an undefined method call.
- Case-sensitive variations of URL rewrites work as expected.
- Resolved a JavaScript syntax error in `bundle.js`.
- The cron-related error `Warning: shell_exec() has been disabled for security reasons...` has been resolved. Magento thanks Stefan Hagspiel for reporting this issue.
- cron no longer runs multiple times unnecessarily.
- Cached static blocks now display properly.

## Magento Open Source 1.9.0.1 Release Notes

Magento Open Source 1.9.0.1 resolved the following issues:

- Customers can no longer apply a coupon from an inactive shopping cart price rule to a purchase.
- Customers using a smartphone or other small viewport can expand subcategories in the web store that uses the new responsive theme.

## Recent Patches

We'd like to draw your attention to several new patches that were recently posted to the Partner Portal and Support Center. These patches deliver important improvements, such as enabling several concurrent administrators to work with the product catalog, and to make it easier to install community-created translation packages.

Details about the patches follow. To install patches, see [How to Get Patches For Magento Commerce](#).

- [General Magento Connect Patches](#)
- [Magento Install Page Displays After SOAP v2 Index Page Refresh](#)
- [How to Get Patches For Magento EE](#)



**Note:** Some of the patches discussed in this section have `EE_1.14.0.1` in the name. These patches were all tested against Open Source 1.9.x as well.

## General Magento Connect Patches

*Patch name:* SUPEE-3941

- When you install a community-created translation package, the translation provided by the package overwrites any existing translations for the same items. This enables you to more easily install packages with translations.
- To improve security, Magento Connect now uses HTTPS by default to download extensions, rather than FTP.
- Extension developers can now create an extensions with a dash character in the name. Merchants can install those extensions without issues.
- Magento administrators who attempt to install an extension with insufficient file system privileges are now informed. Typically, the Magento Admin Panel runs as the web server user. If this user has insufficient privileges to the `your Magento install dir/app/code/community` directory structure, the Magento administrator sees an error message in the Magento Connect Manager.

To set file system permissions appropriately, see [After You Install Magento: Recommended](#)

## Magento Install Page Displays After SOAP v2 Index Page Refresh

*Patch name:* SUPEE-3762. Refreshing the SOAP v2 index page ([http://your-magento-host-name/index.php/api/v2\\_soap/index/](http://your-magento-host-name/index.php/api/v2_soap/index/)) results in all administrators and customers viewing the Magento installation page.

## How to Get Patches For Magento Open Source

This section discusses how to get patches referenced in these Release Notes.

To get patches for Magento Open Source:

1. Log in to [www.magentocommerce.com/download](http://www.magentocommerce.com/download).
2. In the left pane, click **Downloads**.
3. Scroll down to the Magento Open Source Patches section.
4. Follow the prompts on your screen to download a patch for your version of Magento Open Source.
5. Apply the patch as discussed in [How to Apply and Revert Magento Patches](#).

## Magento Open Source 1.9.0.0 Release Notes

See the following sections for information about changes in this release:

- [Highlights](#)
- [Security Enhancements](#)
- [Changes](#)
- [Tax Calculation Fixes](#)
- [Fixes](#)

## Highlights

This section lists the key new features in Magento Open Source 1.9. For more information about these new features, see the [Magento User Guide](#).

- The default theme in Magento Open Source 1.9 uses [Responsive Web Design](#) principles to provide a better experience for users of mobile devices in particular. Benefits include:
  - You can get a tablet and smart phone friendly responsive site in about half the time as before, speeding time to market and freeing up resources for other projects.
  - Your responsive site makes you better able to participate in the fast growing mobile commerce space, gives you the ability to more easily adapt to new opportunities, and is less expensive to maintain. A responsive site also offers potential search engine optimization (SEO) benefits because it uses Google's preferred approach to mobile-optimizing sites.
- Cross-border trade: (Also referred to as *pricing consistency*.) We support European Union (EU) merchants operating across regions and geographies who want to show their customers a single price. Pricing is clean and uncluttered regardless of tax structures and rates that vary from country to country.  
To enable cross-border trade in the Admin Panel, click **System > Configuration > SALES > Tax > Calculation Settings**, option **Enable Cross Border Trade**.
  - Supports [PHP 5.4](#). For more information, see [the PHP changelog](#).
  -

- The Zend Framework has been upgraded to [version 1.12.3](#)
- Checkout improvements:
  - You can capture up to 18% more sales by providing customers access to financing using the Bill Me Later service at no additional cost to you.
  - You can offer your customers a smoother, more streamlined PayPal Express Checkout experience, which tries alternative payment options when a customer's credit card is rejected
  - Improve the PayPal Express checkout experience by eliminating the following steps in the checkout process:
    - The order review page can be enabled or disabled
    - Eliminate the necessity of clicking **Update Order** before **Place Order**

(*Conversion* means helping customers stay interested and complete their purchases.)

## Security Enhancements

- Addressed a potential cross-site scripting (XSS) vulnerability while creating configurable product variants.
- Addressed a potential security issue that could result in displaying information about a different order to a customer.
- Users can no longer change the currency if the payment method PayPal Website Payments Standard is used.
- Removed an .swf file from the Magento distribution because of security issues.
- Improved file system security.
- Enhanced the security of action URLs, such as billing agreements.
- Addressed a potential session fixation vulnerability during checkout.
- Improved the security of the Magento randomness function.

## Changes

- A default setting for configurable and bundled products has changed. When you create a configurable or bundled product in the Admin Panel, click **Manage > Products**. Create a new configurable or bundled product and click the **Design** tab. The default option for **Display Product Options In** has changed to **Product Info Column**.
- The Google Websites Optimizer has been disabled because it has been deprecated by Google. (In earlier Magento Commerce versions, this option was available in the Admin Panel at **System > Configuration > SALES > Google API > Google Website Optimizer**).
- Two new options to prevent "clickjacking" if you run Magento in a **frame** or **iframe**:
  - Enable frames only in the same domain.
  - Enable frames.



**Important:** For security reasons, Magento strongly recommends against running the Magento software in a frame.

The options are available in the Admin Panel at **System > Configuration > ADVANCED > Admin > Security** and are named **Allow Magento Backend to run in frame** and **Allow Magento Frontend to run in frame**.

Enabling the option causes the **X-Frame-Options** request header to be sent.

- FedEx changed their endpoints for sandbox and production to the following:



Production: <https://ws.fedex.com/web-services>

- Sandbox: <https://wsbeta.fedex.com/web-services>

Because of these changes, Magento cannot retrieve shipping rate information or print shipping labels for FedEx unless this fix is applied.

- Open Source 1.9 includes a fix that prevented some Discover credit cards from validating properly. The issue was that certain Discover credit card number ranges were not recognized as being valid. As a result of the fix, all Discover cards should validate properly.



**Important:** This is *not* a security threat. No data has been compromised or misused. It affects only the ability to validate certain credit card number ranges as valid Discover card numbers.

- The default values for two configuration options changed. Both options can be found in the Admin Panel under **System > Configuration > CATALOG > Catalog > Frontend**.

The new default values follow:

- **Products per Page on Grid Allowed Values** is now **12, 24, 36**.
- **Products per Page on Grid Default Value** is now **12**.
- On the New Rule page for shopping cart price rules (**Rule Information** tab page), explanatory text **Usage limit enforced for logged in customers only** was added to the **Uses per customer field**. This is to avoid confusion encountered by some Magento merchants.
- Changes to PayPal Express checkout:
  - Changes to both **Proceed to Checkout** and to **Pay With PayPal**:
    - Customers cannot edit their billing address on the PayPal site—in fact, the billing address does not display on the PayPal site.
    - Magento Open Source uses the customer's PayPal address information, not the address information stored in Magento Open Source. The customer does not need to enter Magento Open Source or Commerce address information.
    - When the customer is redirected to the PayPal site, they can click **Change** next to their payment method to change it.

The Magento administrator can set billing address information in the Admin Panel as follows: **System > Configuration > SALES > Payment Methods**. For any payment method that includes Express Checkout, in Basic Settings, set the value of the **Require Customer's Billing Address** list.

- Changes to **Pay With PayPal** only:
  - If the Magento administrator does not require the customer's billing address, the customer's billing address is set to the customer's PayPal **shipping** address.
  - If Magento requires the customer's billing address, the customer's billing address is set to the customer's PayPal **billing** address.
  - On the PayPal site, the customer can click **Change** next to their shipping address to change it. The customer can choose any shipping address configured with PayPal.
  - Because the customer's configured address in Magento is not used, the customer's shipping address stored in PayPal is never changed.
- Changes to **Proceed to Checkout** only:
  - No **Change** link displays next to a customer's shipping address on the PayPal site.
  - In the event of a shipping address mismatch between Magento and PayPal,

the following message displays when the customer attempts to pay using PayPal:

The address you entered on *store-name* is different than your PayPal preferred shipping address.  
Return to *store-name* if you'd like to change the shipping address

The user can select the **Use as preferred shipping address** checkbox to instruct PayPal to change their shipping address.

## Tax Calculation Fixes

- Fixed price and dynamic price bundled products where the price is configured to include tax display prices correctly regardless of tax settings. (For example, customer's default tax rate is different from the origin tax rate.)
- Resolved a one-cent rounding issue when Fixed Product Tax (FPT) is enabled and the option **Apply Discount to FPT** is set to **Yes**. (These options are available in the Admin Panel by going to **System > Configuration > SALES > Tax > Fixed Product Taxes**.)
- Resolved issues with calculating the credit memo amount when FPT is discounted and the customer purchases more than one item.

## Fixes

Fixes in this release can be divided into the following categories:

- [Web Store and Shopping Cart Fixes](#)
- [Promotional Price Rule Fixes](#)
- [Administrative Ordering, Invoicing, Credit Memo Fixes](#)
- [Import Fixes](#)
- [Payment Method Fixes](#)
- [Other Fixes](#)

### Web Store and Shopping Cart Fixes

- A customer can update quantities of items in their mini shopping cart from their My Account page.
- The Minimum Advertised Price pop-up works properly in the web store. When the customer clicks **Click for price**, the price displays as expected.
- The "customer since" date is correct.
- Switching stores when viewing a product with store-scoped URL keys works as expected.
- Setting **System > Configuration > CATALOG > Inventory**, option **Display Out of Stock Products** to **Yes** no longer causes all products to appear as out of stock.
- Entering accented characters in the zip code field during checkout results in a validation error instead of an exception message.
- Gift card codes are sent only after an item is purchased.
- A customer who attempts to log in as another customer with incorrect credentials is denied.
- Resolved issues with applying a 100% discount to an order.
- Customers are no longer redirected to the home page when they have permission to view a category.

- Discount amount displays correctly for products with custom options.
- Issues with placing PayPal Payments Advanced or PayPal Payflow Link orders using Internet Explorer 9 have been resolved.

## Promotional Price Rule Fixes

The following fixes relate to administering and using shopping cart price rules and catalog price rules:

- Shopping cart price rules apply properly to grouped products.
- Two catalog price rules applied to the same product work properly.
- The setting **Stop Further Rules Processing** is honored.
- A user with read-only privileges in the Admin Panel cannot save changes to a price rule.
- Applying a shopping cart price rule does not display an exception.
- Coupon codes apply only to products eligible for the price rule.

## Administrative Ordering, Invoicing, Credit Memo Fixes

- An administrative user with access to only one website from which a product was deleted no longer sees a stack trace when attempting to create an RMA for that product. In other words, after a customer placed an order for a product on Website1, an administrator with privileges to all websites removes the product. Later, when an administrator with access to only Website1 attempts to create an RMA for the deleted product, that administrator no longer sees an error message; instead, they see an expected **Access Denied** message.
- Resolved an issue with incorrectly calculating the amount of an invoice when some items were discounted by a shopping cart price rule.
- Credit memo amount is calculated correctly when processing a partial invoice with a discount.
- Making comments in a credit memo no longer returns items to stock. (Prerequisite: an administrator set **System > Configuration > CATALOG > Inventory > Product Stock Options**, option **Automatically Return Credit Memo Item to Stock** set to **Yes**.)

## Import Fixes

- A product with a custom attribute set imports successfully.

## Payment Method Fixes

- If guest checkout is disabled, a customer must log in to check out with PayPal Express.
- Eliminated errors in the logs when an administrator clicks **System > Configuration > SALES > Payment Methods**.
- You can now use New Zealand dollars as the base currency with the eWAY Direct payment bridge.
- Store credit is applied correctly when using Website Payments Pro Hosted Solution.
- If the merchant country is Germany (DE), disabled guest checkout for the express checkout method and PayPal Website Payments Standard.

## Other Fixes

-

- Categories on the web store now display with spaces between category names for cached and non-cached pages.
- A customer can now initiate a return from the web store.
- An administrative user can subscribe to low stock RSS feeds without errors.
- Category URLs work as expected, regardless of the setting of **Create Custom Redirect for old URL** for the category's URL key.
- Setting `allow_url_fopen = Off` in `php.ini` has no effect on the CMS WYSIWYG editor.
- No fatal error displays when a role-restricted user previews a newsletter in the Admin Panel.
- Google Sitemap files now include the `.html` suffix for category and product URLs.
- Customers can use advanced search on your web store if Magento EE is configured to use the default MySQL Fulltext search engine and the server uses MySQL 5.6.
- A role-restricted user can preview a newsletter in the Admin Panel to which the user has privileges.
- After synchronizing media files with the database, `media/customer/.htaccess` is present with the correct data. (Prerequisite: an administrator set **System > Configuration > ADVANCED > System > Storage Configuration for Media** set to **Database**).
- cron now restarts indexers if they previously failed to run.
- You can save changes to a category that has more than 1,000 products.
- Deactivating one of several banners no longer causes exceptions in `system.log`.
- Resolved issues with the WSDL cache.
- Improved the efficiency of product searches.
- Resolved issues with the DHL International shipping method.
- Resolved 404 (Not Found) errors in layered navigation.
- Resolved a SQL error when attempting to assign a bundled product to another website.
- Rules-based product relations perform as expected after being saved.
- Resolved an issue with sending duplicate Content-Type headers when using `mod_fastcgi` with the Apache web server.

## Open Source Software Licensing Agreements

Some versions of Magento Open Source use open source software licensing. Following are license agreements for that software.

**Touch punch:** This code is dual licensed under the MIT or GPL Version 2 licenses and is therefore free to use, modify and/or distribute, but if you include Touch Punch in other software packages or plugins, please include an attribution to the original software and a link to this [Touch Punch website](#).

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento Commerce Release Notes (1.14 and later)

## Contents

These Release Notes contain the following information:

- [Important Upgrade Information](#)
- [Magento Commerce 1.14.4.5 Release Notes](#)
- [Magento Commerce 1.14.4.4 Release Notes](#)
- [Magento Commerce 1.14.4.3 Release Notes](#)
- [Magento Commerce 1.14.4.2 Release Notes](#)
- [Magento Commerce 1.14.4.1 Release Notes](#)
- [Magento Commerce 1.14.4.0 Release Notes](#)
- [Magento Commerce 1.14.3.10 Release Notes](#)
- [Magento Commerce 1.14.3.9 Release Notes](#)
- [Magento Commerce 1.14.3.8 Release Notes](#)
- [Magento Commerce 1.14.3.7 Release Notes](#)
- [Magento Commerce 1.14.3.6 Release Notes](#)
- [Magento Commerce 1.14.3.5 Release Notes](#)
- [Magento Commerce 1.14.3.4 Release Notes](#)
- [Magento Commerce 1.14.3.3 Release Notes](#)

- [Magento Commerce 1.14.3.2 Release Notes](#)
- [Magento Commerce 1.14.3.1 Release Notes](#)
- [Magento Commerce 1.14.3.0 Release Notes](#)
- [Magento Commerce 1.14.2.0 Release Notes](#)
- [Magento Commerce 1.14.1.0 Release Notes](#)
- [Magento Commerce 1.14.0.1 Release Notes](#)
- [Recent Patches](#)
- [Magento Commerce 1.14.0.0 Release Notes](#)
- [Open Source Software Licensing Agreements](#)

## Important Upgrade Information



**Important:** Use Magento Commerce 1.14.3.0 or later for *all new Magento Commerce installations and upgrades* to get the latest fixes, features, and security updates.

## Magento Commerce 1.14.4.5 Release Notes

This version (or patch SUPEE-11314, which applies to older versions of Magento) provides resolution of multiple critical security issues. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

We recommend upgrading your Magento store to this latest version. See [Magento | APSB20-22](#) for a comprehensive discussion of these issues.

## Magento Commerce 1.14.4.4 Release Notes

This version (or patch SUPEE-11295, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

We recommend upgrading your Magento store to this latest version. See [Magento | APSB20-02](#) for a comprehensive discussion of these issues.

## Fixed issues and enhancements

- The **Disable** button present when you run the compiler from **Admin > System > Tools > Compiler** is now enabled as expected. Previously, when you clicked the **Disable** button, it did not change state.

## Magento Commerce 1.14.4.3 Release Notes

This version (or patch SUPEE-11219, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#)

for a comprehensive discussion of these issues.

## Fixed issues and enhancements

- WebserviceX has been removed from the Magento 1.x code base.
- This release adds two new currency services for currency rate import: [CurrencyConverterAPI](#) and [FixerIO](#)

## Known issue

This release includes a fix for a security vulnerability that potentially allowed changes to protected store settings. As a result, extensions or customizations that depend on saving configuration fields that are not defined in `system.xml` files may no longer work correctly.

## Magento Commerce 1.14.4.2 Release Notes

This version (or patch SUPEE-11155, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

**Note:** We are aware of the incompatibilities between patch SUPEE-11155 and the PHP 7.2 support patch and are currently working on a new version of SUPEE-11155 that resolves these incompatibilities. See [Security Patch SUPEE-1155 - Possible issues?](#) for a community-driven discussion on issues and solutions related to this SUPEE. Check these release notes and the [Magento Security Center](#) for updates on the availability on the new patch.

We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

## Fixed issues and enhancements

- The Magento logging feature now works as expected after the SUPEE-11086 patch is installed. Previously, after application of this patch, Magento could only write only to a file that already existed on the server, and did not create new log files.
- Magento 1.14.4.0 and the PHP7.2 support patch now include the same files as expected. The previous version of the patch did not include the following three files, which were included in Magento 1.14.4.0. Magento 1.14.4.0:  
`lib/phpseclib/PHP/Compat/Function/array_fill.php`,  
`lib/phpseclib/PHP/Compat/Function/bcpowmod.php`, and  
`lib/phpseclib/PHP/Compat/Function/str_split.php`.

## Known issues

The extensive security enhancements we've included to this release have resulted in the following changes to Magento behavior:

- You can no longer upload files with the extension `.swf` to the WYSIWYG editor.
- Third-party checkout extensions and closed security cases will either not work securely or will not work at all.
- The Authorize.net Direct Post module has been enhanced to support the replacement of Authorize.net's MD5-based hash with a (SHA-512) signature key. Authorize.net will no longer support implementations using the MD5-based hash as of June 28, 2019. You will need to update your signature key after upgrading to this version of Magento. For information about updating your signature key, see the Get a New Signature Key discussion in the [Update Authorize.Net Direct Post from MD5 to SHA-512](#) help article. Note that although this help article

describes how to install the earlier patch, merchants upgrading to this release of Magento are not applying the patch and should consult only the Get a New Signature Key discussion. If you've applied the patch to your Magento installation while running an earlier version of Magento, uninstall the Update Authorize.Net Direct Post from MD5 to SHA-512 patch before upgrading to this release.

- You can no longer preview JavaScript in a newsletter template in the Admin.
- Sitemap names cannot exceed 32 characters.

## Magento Commerce 1.14.4.1 Release Notes

This version (or patch SUPEE-11086, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These security enhancements help close cross-site scripting, arbitrary code execution, and sensitive data disclosure vulnerabilities as well as other security issues.

We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

**Note:** Magento's implementation of the Authorize.Net Direct Post payment method currently uses MD5-based hash for all M1 and M2 installations. As of June 28, 2019, Authorize.Net will stop supporting MD5-based hash usage.

This will result in Magento merchants not being able to use Authorize.Net Direct Post to process payments. To avoid disruption and to continue processing payments, merchants must apply a patch provided by Magento and add a Signature Key (SHA-512) in the Magento Admin configuration settings. Magento released this patch in late February to address this issue on pre-2.3.1 installations of Magento. See [Update Authorize.Net Direct Post from MD5 to SHA-512](#).

Information about the deprecation of Authorize.Net Direct Post can be found in [Authorize.net Direct Post \(Deprecated\)](#).

## Fixed issues and enhancements

- Google Image Charts has been deprecated and replaced by [Image-Charts](#) for dashboard charts.
- Layered navigation now works as expected when full page cache and block caching are enabled. Previously, you could not clear layered navigation filters when these features were enabled.
- Errors caused by problematic PHP error logging have been resolved. Previously, Magento displayed excessive and unnecessary 404 errors.
- Magento now displays the following message when an invalid character is used, **Attribute code is invalid. Please use only letters (a-z), numbers (0-9) or underscore(\_) in this field, first character should be a letter. Do not use "event" for an attribute code.** Previously, Magento did not flag invalid attribute codes.
- You can now add to the cart products with custom options for which the **custom option** checkbox has not been checked. Previously, Magento did not add the product to the cart, and displayed this message, **Cannot add the item to shopping cart.**
- URL redirects for products now work as expected. Previously, when you selected a product from the Category page and Add URL Redirect has been enabled, Magento redirected users to URL Redirect Information and threw this error, **exception 'Mage\_Core\_Exception' with message 'Invalid block type: Mage\_Adminhtml\_Block\_Empty\_Edit\_Form' in app/Mage.php:580**
- Magento now displays payment information during the confirmation step of check out and

successfully processes an order when inline translation is enabled. Previously, Magento did not display this payment information during check out, and the order was not completed.

- You can now create a staging website when development mode is enabled. Previously, Magento threw an error after you added a website from System > Content Staging > Staging Websites.
- You can now successfully delete a website by clicking **Delete Website** as expected. Previously, when you clicked this button, Magento threw a fatal error.
- You can now add a banner by clicking **Add Banner** from the Admin. Previously, Magento threw an error when you clicked this button.
- Magento no longer throws an `Undefined index: is_recurring` error when you try to save a product when deploying Magento with development mode enabled.

## Known issue

- If there were any custom widgets created by usage of helper methods in layout updates, they will no longer work.

## Magento Commerce 1.14.4.0 Release Notes

This version (or patch SUPEE-10975, which applies to older versions of Magento) provides resolution of multiple critical security issues and functional fixes. These critical security issues include remote code execution (RCE), cross-site scripting (XSS), and cross-site request forgery (CSRF) issues. This release also provides support for PHP 7.2.

We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

Note that after updating to this release, third-party modules that depend upon Magento core backup functionality will no longer work. Alternatively, you can use one of these two methods to enable database backups:

- Change `false` to `true` in the **config > modules > Mage\_Backup > active** setting of `app/etc/modules/Mage_All.xml`.
- From the Admin, change the **System > Configuration > Advanced > Disable Modules Output > Mage\_Backup** from `disabled` to `enabled`.

## Fixed issues and enhancements

- This release provides support for PHP 7.2.
- We've removed the CC module. As a result, third-party modules that depend upon either the `ccsave` method or the `xmlconnect` method will not work as expected. Third-party themes that implement `ccsave` will not work as expected, either.
- The Magento logo has been updated throughout the code base.
- The **Continue** button now works as expected on the Payments step of checkout when paying with the PayPal payment method.
- Google Tag Manager now logs sales information in Google Analytics as expected.
- The product export CSV file now contains columns for super attributes.
- Magento no longer throws an error when a customer accesses their shopping cart after items in their cart have been removed due to a timeout. Previously, Magento displayed this error, `'Notice: Undefined variable: freePackageValue in`

- Clicking on a configurable product's swatch on the product list page now updates product price as expected.
- Customers can now successfully add a grouped product to their shopping cart when category permissions are enabled. Previously, Magento did not add the product to the cart, but instead displayed a descriptive error message.
- Magento no longer displays incorrect prices on the storefront after a failure of the enterprise refresh index.
- We've resolved issues in the indexing locking mechanism that previously resulted in Magento throwing an exception after indexing completed.
- Magento no longer throws a fatal error when a merchant uses an already reserved word to name a product attribute.
- Magento now adds the correct sales tax to orders being shipped to U.S. addresses that use zip codes with the optional four-digit suffix (for example, 73365-1234). Previously, the Tax rule triggered a failure if the U.S. zip code that had this optional four-digit suffix.
- Magento now displays all products on a production website that were edited by a role-restricted user on the associated staging website.
- We've resolved an issue that caused Target Rules to throw an exception when a customer opened a product view page.

## Magento Commerce 1.14.3.10 Release Notes

This version (or patch SUPEE-10888, which applies to older versions of Magento) provides resolution of multiple critical security issues. These critical security issues include remote cross-site scripting and cross-site request forgery issues. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

**Note:** With this release, Magento is announcing the following support policy: Magento will provide software support through June 2020 for Magento Commerce 1.x. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our Magento Software Lifecycle Policy [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

### Known issue

You cannot re-send the password for new customers who created their account during checkout.

## Magento Commerce 1.14.3.9 Release Notes

This version (or patch SUPEE-10752, which applies to older versions of Magento) provides resolution of multiple critical security issues. These critical security issues include remote code execution, cross-site scripting, and cross-site request forgery issues. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

### Fixed issues and enhancements

- Magento no longer performs unnecessary write operations on the `core_url_rewrite` table.
- Customers can now successfully register during checkout without being unexpectedly logged out.
-

Incorrect escaping in the `cron.sh` file no longer prevents cron jobs from running in parallel as expected.

- Magento now cleans session data as expected after a customer logs out.

## Known issue

If your custom code or extension is using `Zend/Filter/PregReplace.php` with the modifier `e`, it will now return an error due to possible RCE issues. See APPSEC-2029 in [Magento Security Center](#) for more information.

## Magento Commerce 1.14.3.8 Release Notes

This version (or patch SUPEE-10570, which applies to older versions of Magento) provides resolution of multiple critical security issues. These critical security issues include remote code execution, cross-site scripting, and cross-site request forgery issues. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

## Fixed issues and enhancements

- **Changed Magento Admin to support recent USPS shipping changes.** On February 23, 2018, USPS removed APIs that support the creation of shipping labels without postage. In response, we've removed this functionality from the Magento Admin. Consequently, you cannot create and print shipping labels that do not have postage applied.
- **Updated copyright to 2018.**

## Known issues

These two known issues are associated with the use of HTML tags within a product's SKU attribute:

- If you try to import products that contain HTML tags in the SKU attribute, Magento displays this error at the data validation stage (that is, when you click **Check data**): `Invalid value in SKU column. HTML tags are not allowed.`
- If you try to create or edit a product in the Admin panel and the product's SKU attribute value contains HTML tags, Magento throws this error when you try to save the product: `HTML tags are not allowed in SKU attribute.`

## Magento Commerce 1.14.3.7 Release Notes

This patch (SUPEE-10415) provides resolution of multiple critical security issues. These critical security issues include remote code execution, cross-site scripting, and cross-site request forgery issues. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

## Fixed issues

- Magento no longer displays the "Invalid Secret Key. Please refresh the page." message when a user loads the Admin.
- The one-page checkout page now displays the following message when a customer checks out an order for which no amount is due: **No payment information required.** Magento versions prior to 1.14.3.3 included this message, but it was missing from v1.14.3.3.
- We've fixed a typo in the patch header information. (`autocomplete="new-pawwsord"` is now `autocomplete="new-password".`)

## Known issue

**Issue:** Magento displays a "404: Page Not Found" error from the `errors/` directory after upgrading to SUPEE-10415. This issue occurs only in Magento installations that run certain third-party extensions.

**Description:** Magento is not properly logging PHP warnings that occur early during page initialization. Instead, of logging the error and continuing operation, Magento generates a 404 page. (Previously, Magento logged these warnings in the `system.log` file, and execution would continue as usual.)

**Workaround:** Confirm that there are no PHP warnings generated by any of the extensions or customizations.

## Notes

- We no longer support custom file extensions for `Mage::log()`. Supported file extensions include `.log`, `.txt`, `.html`, `.csv`. For more information, navigate to **Developers > Log Settings** from the Admin. Magento displays this comment: **Logging from Mage::log(). File is located in /var/log. Allowed file extensions: log, txt, html, csv.**
- Passwords for new users are now limited to 256 characters. If a new user enters a password that exceeds 256 characters, Magento displays this message: **Please enter a password with at most 256 characters.**

## Magento Commerce 1.14.3.6 Release Notes

This patch (SUPEE-10266) provides resolution of multiple critical security issues and several functional fixes. These critical security issues include remote code execution, cross-site scripting, and cross-site request forgery issues. We recommend upgrading your Magento store to this latest version. See [Magento Security Center](#) for a comprehensive discussion of these issues.

SUPEE-10266 includes a fix for MPERF-9685. This fix is not included in release 1.14.3.6. However, in some cases, SUPEE-10266 can cause issues in the checkout process. Specifically, if a customer enables the Add gift options checkbox during checkout, the checkout process will not progress beyond the payments step.

**Note:** We've released patch SUPEE-10348 to address issues with checkout that some users experienced after applying SUPEE-10266. SUPEE-10348 must be applied on top of SUPEE-10266. Note that if you are experiencing issues in checkout when using gift options, SUPEE-10348 should resolve these issues.

If you are currently affected by this issue, you can workaround this issue by restore these two files to the pre-patch versions:

`app/design/frontend/enterprise/default/template/giftcardaccount/onepage/payment/scripts.phtml`

`app/design/frontend/rwd/enterprise/template/giftcardaccount/onepage/payment/scripts.phtml`

We've also fixed an issue where uploaded images were twice their original size after you applied SUPEE-9767 v2.

## Magento Commerce 1.14.3.5 Release Notes

We have skipped release 1.14.3.5.

# Magento Commerce 1.14.3.4 Release Notes

This patch (SUPEE-9767 version 2) addresses both security and functional issues discovered when using the SUPEE-9767 patch. We recommend upgrading your Magento store to this latest version.

Here are your upgrade options:

- Upgrade to Magento Commerce 1.14.3.4. You don't need to revert any patches or install SUPEE-9767 version 2 as version 2 is already included in the 1.14.3.4 release.
- If you've already installed SUPEE-9767 version 1, you can either:
  - Upgrade to Magento Commerce 1.14.3.4, or
  - revert SUPEE-9767 version 1, then install SUPEE-9767 version 2.
- If you haven't installed SUPEE-9767 version 1, you can either:
  - Upgrade to Magento Commerce 1.14.3.4, or
  - install SUPEE-9767 version 2.

See [Magento Security Center](#) for a comprehensive discussion of these security issues.

This release also provides support for the following functional issues:

## General fixes

- We've added a missing `form_key` to the enterprise shipping template (`app/design/frontend/enterprise/default/template/checkout/onepage/shipping.phtml`).
- We've restored missing `strip_tags` functionality in the checkout JavaScript.
- We've changed how Magento validates form keys during the generic five-step checkout process. Previously, customer registration failed during standard checkout processing if form key authentication was enabled.
- You can now successfully add a product to the shopping cart when FPC (Full Page Cache) is enabled. Previously, when FPC was enabled, adding a product to the cart resulted in an empty cart.
- Magento now displays the `Allow_symlinks` message in the Admin message area as expected.
- Magento now preserves the background transparency of uploaded images as expected. Previously, transparency was lost after the image was uploaded, resulting in an unusable image.
- You can now use Checkout with Multiple Addresses when checkout form validation is enabled.

## Installation

- The `Allow symlinks` option is now disabled during installation or upgrade processes. Previously, when you changed the `Allow symlinks` setting to true in the database before upgrading and then installing the patch, this option remained enabled, but you could no longer access it from the Admin panel.

This patch is available from [Magento Tech Resources](#).

# Magento Commerce 1.14.3.3 Release Notes

This patch provides resolution of multiple critical security issues. These critical issues include remote code execution for authenticated Admin users, access control bypass, and cross-site request forgery issues. See [Magento Security Center](#) for a comprehensive discussion of these issues.

This release also provides support for the following issue:

Support for PayPal's update to its Instant Payment Notification (IPN) server URL. PayPal provides more information about this feature in [IPN Verification Postback to HTTPS Microsite](#). This update is essential for retaining uninterrupted service after June 30.

SUPEE-8167, an older patch that also contains this fix, was added on May 8, 2017, and is available from [Magento Tech Resources](#).

## Known Issues

This patch/release has known issues. Please see [SUPEE-9767](#) for updates.

**Note:** Before applying this patch or updating to this release, disable the **Symlinks** setting in **System > Configuration > Advanced > Developer > Enable Symlinks**. If the **Symlinks** setting is enabled, it will override your configuration file settings. If that override occurs, you will need to directly modify the database to change those settings.

## Magento Commerce 1.14.3.2 Release Notes

This patch addresses the following issues:

- Removal of vulnerability with the Zend framework `Zend_Mail` library. For more information, see [Magento Security Center](#).
- Updated the copyright year to 2017.

## Magento Commerce 1.14.3.1 Release Notes

This patch addresses the following issues:

- We restored the old tax calculation algorithm for shipping charges. The patch to apply new calculation will be available on request.
- Resolved an issue with setting the session lifetime to 0.
- The monthly cron job that cleans up the table that contains IP addresses and passwords runs properly.
- All configurable product images are imported.
- You no longer get an exception due to an undefined `addCrumb()` method call.
- Resolved the error `Notice: Undefined index: session_expire_timestamp` when accessing the storefront.
- Values for drop-down label values are saved correctly.
- The "Price as configured" for bundle products displays correctly in the shopping cart.
- Auto-generated passwords are sent to new customers as expected.
- The method `Mage_Api_Model_Server_Handler_Abstract::processingMethodResult()` accepts scalar and array values.
- The default MySQL Full-Text search works as expected; it no longer returns all products.
- Prevented a potential Cross-Site Request Forgery (CSRF) vulnerability by changing the form

key when a customer signs out of the storefront.

- Prevented a potential [Cross-Site Scripting \(XSS\)](#) vulnerability when adding a category.
- Catalog price rules return the correct price.
- Indexers now update all products instead of skipping the last product updated.



**Note:** You currently *cannot* upgrade to this version using Magento Connect Manager. We expect to resolve this issue soon.

## Magento Commerce 1.14.3.0 Release Notes

See the following sections for information about this release:

- [Highlights](#)
- [Security Enhancements](#)
- [Check for .swf File After Upgrade](#)
- [Backward-Incompatible Changes](#)
- [Fixes](#)

### Highlights

Magento Commerce 1.14.3 delivers more than 120 quality improvements, as well as support for PHP 5.6 in addition to PHP 5.4 and 5.5.

### Security Enhancements

We addressed the following security issues in this release:

- [General security enhancements](#)
- [Password enhancements](#)

#### General security enhancements

For more information about these security enhancements, see [our Security Center article](#).

Patches for major security issues in earlier versions of the Magento software are available in the Magento Commerce customer portal.

Details about security fixes:

- Resolved a potential SQL injection (Zend Framework issue)
- Resolved a cache poisoning issue
- We now provide better protection against path exploits.
- Resolved a potential [cross-site scripting \(XSS\)](#) vulnerability when adding a category.
- Resolved a potential XSS vulnerability that affected the Magento server's request URI.
- Resolved a potential XSS vulnerability in invitations.
- You can no longer cause out-of-memory errors on the Magento server by flooding it with images that have incorrect dimensions.
- The Magento Admin Panel login page now renders in HTTPS if you configured the Magento server for HTTPS.

- We added the **nosniff** header to our .htaccess files.
- Magento no longer uses Adobe Flash for uploads.
- Fixed several potential issues indicated by static code scans.
- Resolved a potential man-in-the-middle vulnerability.
- Resolved a potential PHP security vulnerability.
- An administrative user is no longer able to create a potential security vulnerability that used the block cache.
- Resolved a potential cross-site request forgery (CSRF) vulnerability involving the wishlist.
- Resolved a potential remote code execution exploit.
- It is no longer possible to log in to a store as an existing customer using only an e-mail address.

## Password enhancements

- A user can reset a password only after receiving an e-mail. In addition, we introduced the following configuration settings:
  - Limit the number of forgotten password requests from one IP address to five times per hour.
  - Limit the number of forgotten password requests from one e-mail address to five times per 24 hours.
  - Limit the number of forgotten password requests to no more than once every 10 minutes per e-mail address.
- The forgot password link expires after the first use or two hours (by default).
- When a user changes their e-mail address, they are required to provide their password and to acknowledge the change from the previous address.
- We now ignore leading and trailing spaces in a user's password.
- The new customer e-mail now includes the customer's password.
- Resetting a password using a password recovery e-mail succeeds.

## Check for .swf Files After Upgrade

If you upgraded to Magento Commerce 1.14.3 after applying the [SUPEE-8788 patch](#), make sure the following files have been deleted:

```
skin/adminhtml/default/default/media/flex.swf
skin/adminhtml/default/default/media/uploader.swf
skin/adminhtml/default/default/media/uploaderSingle.swf
```

If the files are present, delete them to avoid a potential security exploit. As of Magento Commerce 1.14.0.0, we no longer distribute .swf files with the Magento software.

## Backward-Incompatible Changes

The following backward-incompatible changes were made in this release:

**Mage\_Adminhtml\_Block\_Cms\_Wysiwyg\_Images\_Content\_Uploader**: Parent class was removed.

`Mage_Uploader_Model_Config_Abstract`: Overrides the magic method `__call` and its behavior can be inconsistent. For example:

```
->setData('underscore_key', 1)
->getUnderscoreKey() //null
```

## Fixes

The following sections discuss other fixes in this release:

- [Tax Calculation Fixes](#)
- [Shopping cart and checkout fixes](#)
- [Catalog fixes](#)
- [Price rule fixes](#)
- [Visual Merchandiser fixes](#)
- [Configurable swatches fixes](#)
- [Import/export fixes](#)
- [Indexer fixes](#)
- [Other fixes](#)

### Tax Calculation Fixes

- The subtotal including tax on an invoice is calculated correctly.

### Shopping cart and checkout fixes

- One product displays one time in a cart even if the product was added once as a guest and another time as a logged-in user.
- Bundled products now display properly in the mini cart as well as the shopping cart.
- Moving a configurable product to a shopping cart in the Admin Panel functions normally.
- Shipping discount coupons are now based correctly on a customer's shipping address.
- First Class Mail letter now displays as a shipping option in the shopping cart.
- You can now pay for a product using both store credit and reward points.
- An exception no longer displays when a customer uses a gift card in an invalid transaction (such as an incorrect payment card number).
- We added validation so a special price must be less than the actual price.
- Exceptions no longer display when a customer checks out.
- Fixed a programming issue that prevented serializing and unserializing values in the shopping cart.
- Magento recovers from payment processor unavailability properly; the customer is charged and the item is shipped.
- You can no longer order an empty product; that is, a product with no options.

## Catalog fixes

- A configurable product with decimal quantity less 1 now displays the proper quantity in the catalog.
- Configurable products are now sorted by attribute, not by product ID.
- Errors no longer display when you use `Mage_Catalog_Block_Product_List` on a product detail page.
- Removed the undefined variable `where` from  
`app/code/core/Mage/CatalogSearch/Model/Resource/Fulltext.php`.

## Price rule fixes

- A catalog price rule that targets a bundled product by percentage calculates the price properly.
- A shopping cart price rule that includes tax now calculates properly.
- With the flat product catalog enabled, a catalog price rule with multi-select attributes works properly.
- Errors no longer display when two users add a product at the same time. Magento thanks Babenko eCommerce for contributing this fix.
- You can now add configurable products to the shopping cart after configuring a shopping cart rule.

## Visual Merchandiser fixes

- We bundled the following fixes in a patch:
  - The value of **Attribute** in the Smart Category section displays the entire category name.
  - We reduced the frequency of rebuilding the Visual Merchandiser index to improve efficiency and performance.
  - Duplicate products no longer display when you choose to hide not-visible products.
  - Reapplied patches for various functions like indexing.
- Improved the display of the out of items message.

## Configurable swatches fixes

- Fixed a memory leak in the configurable swatches module.
- Configurable swatches for out-of-stock products now display consistently in layered navigation, the category view page, and the product view page.
- Configurable swatches work properly even if there is no image (before the fix, a JavaScript error was thrown).
- Resolved performance issues.
- Swatch images for configurable products display properly.

## Import/export fixes

- We bundled the following fixes in a patch:

- Exporting a large number of products no longer results in an out-of-memory error.
- You can import into multiple stores if some stores are set to be replaced.
- Re-importing customers that have a multi-select attribute preserves the attribute.
- File uploads are processed properly.
- Fixed broken help links in the Magento Admin Panel.
- Importing products no longer consumes an excessive amount of memory.
- Coupon reports exported as .csv now display the correct totals.

## Indexer fixes

- With flat category tables enabled, reindexing no longer removes the category class tag.
- Resolved errors with the Product Flat Index not completely indexing a large number of changes.
- All indexes now reindex when set to update when scheduled.
- Improved performance of the category indexers. Magento thanks Vaimo for contributing this fix.
- Categories saved with a / character as the suffix display properly.

## Other fixes

- Applied United States Postal Service API changes for [January 17, 2016](#).
- Default variable values now save normally.
- The WYSIWYG editor handles XHTML tags like cellpadding and cellspacing properly.
- The configuration setting **Allow HTML Tags on Frontend** is honored.
- Orders created using the Magento Admin now display on the Orders and Returns page on the storefront.
- The option to merge Cascading Stylesheets (CSS) and JavaScript now works properly with a responsive web design (RWD) theme.
- On a mobile device when the Magento storefront uses an RWD theme, the Filter bar displays one time only.
- The Magento Connect Manager downloader's .htaccess file is no longer overwritten when the downloader component is updated.
- The configuration cache is no longer corrupted under heavy load.
- Order update e-mails are sent only once.
- A SOAP API call to /api/soap/?wsdl returns normally.
- A value that contains special characters is handled without errors by the SOAP API.
- Fixed the untranslatable base/default/template/sales/guest/form.phtml template.
- Magento now stores two-digit birth years properly (for example, 80 is stored as 1980).
- HTTP 200 (OK) status codes are returned for pages after a session expires.

- You can view a disabled product without errors if compilation is enabled.
- A Value Added Tax (VAT) ID now validates properly. If the customer specifies an invalid ID, the customer is notified they will be charged VAT tax.
- Listing shipments no longer displays an exception.
- You can filter associated products for a group product without errors.
- When you manage product attributes, selecting an action from **Actions** works properly.
- You can now add a configurable product by SKU to an order using the Admin Panel.
- You can now save a product's weight attribute.
- You can now save changes to a CMS page hierarchy when hierarchy metadata is disabled.
- You can now save a banner after upgrading.
- Using a Portable Network Graphics (.png) image on a CMS page no longer results in a `HEADERS_ALREADY_SENT` message to be logged.
- Fixed an exception related to an unknown database table.
- Fixed an issue with JavaScript merging.
- You can now print 10 or more shipping labels without issues.
- A PHP notice no longer occurs when you log the Magento Admin Panel IP address in the event log.
- A SQL error no longer displays when you create a new floating point product attribute programmatically.
- Added a missing image to the codebase.
- The expression `Mage::getModel('core/variable')->addValuesToResult()` returning a collection with column `plain_value` and `html_value` now returns a collection with columns `plain_value` and `html_value`.
- Payment no longer results in the exception `ERR (3): Notice: Undefined offset: 1 in app/code/core/Mage/Sales/Model/Order.php on line 1258`.
- The correct telephone number displays in transactional e-mails. We changed the variable `phone` to `store_phone`. Magento thanks Classy Llama Studios for contributing this fix.
- The Google sitemap now lists store URLs properly.
- Implemented search query caching, which speeds up search results.
- After a customer submits an order, the following error should not display: `SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'EC00000148' for key 'UNQ_SALES_FLAT_ORDER_INCREMENT_ID'`.
- With flat category enabled, you no longer see errors due to an undefined method call.
- Case-sensitive variations of URL rewrites work as expected.
- Resolved a JavaScript syntax error in `bundle.js`.
- The cron-related error `Warning: shell_exec() has been disabled for security reasons...` has been resolved. Magento thanks Stefan Hagspiel for reporting this issue.
- cron no longer runs multiple times unnecessarily.

- Cached static blocks now display properly.

## Magento Commerce 1.14.2.0 Release Notes

Magento Commerce 1.14.2.0 Release Notes are in the [User Guide](#).

## Magento Commerce 1.14.1.0 Release Notes

Magento Commerce 1.14.1.0 Release Notes are in the [User Guide](#).

## Magento Commerce 1.14.0.1 Release Notes

Magento Commerce 1.14.0.1 resolves the following issues:

- Customers can no longer apply a coupon from an inactive shopping cart price rule to a purchase.
- Customers using a smartphone or other small viewport can expand subcategories in the web store that uses the new responsive theme.

## Recent Patches



**Note:** The patches discussed in this section are built in to Commerce 1.14.1; you need to get them only if you're running an earlier Commerce version.

We'd like to draw your attention to several new patches that were recently posted to the Partner Portal and Support Center. These patches deliver important improvements, such as enabling several concurrent administrators to work with the product catalog, and to make it easier to install community-created translation packages.

Details about the patches follow. To install these patches, see [How to Get Patches For Magento Commerce](#).

- [General Magento Connect Patches](#)
- [Magento Install Page Displays After SOAP v2 Index Page Refresh](#)
- [Multiple Simultaneous Magento Administrators](#)
- [How to Get Patches For Magento Commerce](#)

## General Magento Connect Patches

*Patch name:* PATCH\_SUPEE-3941\_EE\_1.14.0.1\_v1-2014-08-12-12-10-06.sh

- When you install a community-created translation package, the translation provided by the package overwrites any existing translations for the same items. This enables you to more easily install packages with translations.
- To improve security, Magento Connect now uses HTTPS by default to download extensions, rather than FTP.
- Extension developers can now create an extensions with a dash character in the name. Merchants can install those extensions without issues.
- Magento administrators who attempt to install an extension with insufficient file system privileges are now informed. Typically, the Magento Admin Panel runs as the web server user. If this user has insufficient privileges to the *your Magento install dir/app/code/community* directory structure, the Magento administrator sees an error message in the Magento Connect

Manager.

To set file system permissions appropriately, see [After You Install Magento: Recommended File System Ownership and Privileges](#).

## Magento Install Page Displays After SOAP v2 Index Page Refresh

*Patch name:* PATCH\_SUPEE-3762\_EE\_1.14.0.1\_v1.sh. Refreshing the SOAP v2 index page (`http://your-magento-host-name/index.php/api/v2_soap/index/`) results in all administrators and customers viewing the Magento installation page.

## Multiple Simultaneous Magento Administrators

*Patch name:* PATCH\_SUPEE-3819\_EE\_1.14.0.1\_v1.sh. Multiple Magento administrators can simultaneously add new products; or edit descriptions, edit prices, or edit stock quantities of existing products without causing deadlocks, key violations, or critical data errors. Together with applying the patch, you must set all indexers to Update when scheduled as follows:

1. Log in to the Magento Admin Panel as an administrator.
2. Click **System > Configuration**.
3. In the left navigation bar, from the ADVANCED group, click **Index Management**.
4. Expand **Indexing Options**.
5. From each list, click **Update when scheduled**.
6. Click **Save Config** in the upper right corner of the page.

## How to Get Patches For Magento Commerce

This section discusses how to get patches referenced in these Release Notes. Magento has other patches available from the Commerce support portal and the [partner portal](#); you can use the following instructions to install any of those patches as well.

To get patches for Magento Commerce:

1. Log in to [www.magentocommerce.com](http://www.magentocommerce.com).
2. In the left pane, click **Downloads**.
3. In the right pane, click **Magento Commerce**.
4. Follow the prompts on your screen to download a patch for your version of EE.
5. Apply the patch as discussed in [How to Apply and Revert Magento Patches](#).

## Magento Commerce 1.14.0.0 Release Notes

See the following sections for information about changes in this release:

- [Highlights](#)
- [Security Enhancements](#)
- [Changes](#)
- [Tax Calculation Fixes](#)
- [Fixes](#)

## Highlights

This section lists the key new features in Magento Commerce 1.14. For more information about these new features, see the [Magento User Guide](#).

- The default theme in Magento Commerce 1.14 uses [Responsive Web Design](#) principles to provide a better experience for users of mobile devices in particular. Benefits include:
  - You can get a tablet and smart phone friendly responsive site in about half the time as before, speeding time to market and freeing up resources for other projects.
  - Your responsive site makes you better able to participate in the fast growing mobile commerce space, gives you the ability to more easily adapt to new opportunities, and is less expensive to maintain. A responsive site also offers potential search engine optimization (SEO) benefits because it uses Google's preferred approach to mobile-optimizing sites.
- Magento Commerce 1.14 now supports Solr versions up to 3.6.2 natively (that is, without a patch). Catalog indexing happens efficiently and automatically in the background, with no manual intervention required, resulting in better administrative performance.  
For more information about using Solr with Magento Commerce, see the [Magento User Guide](#).

**Note:** If you're using the Solr search engine with Commerce versions 1.13.1 or earlier, you must perform an additional step during upgrade due to the fact that the Solr schema changes in Commerce 1.14. You must copy two files to your Solr server—`schema.xml` and `solrconfig.xml`. For details, see [the section on upgrading Solr in the Magento upgrade guide](#).

- Cross-border trade: (Also referred to as *pricing consistency*.) We support European Union (EU) merchants operating across regions and geographies who want to show their customers a single price. Pricing is clean and uncluttered regardless of tax structures and rates that vary from country to country.

To enable cross-border trade in the Admin Panel, click **System > Configuration > SALES > Tax > Calculation Settings**, option **Enable Cross Border Trade**.

- Supports [PHP 5.4](#). For more information, see [the PHP changelog](#).
- The Zend Framework has been upgraded to [version 1.12.3](#).
- Checkout improvements:
  - You can capture up to 18% more sales by providing customers access to financing using the Bill Me Later service at no additional cost to you.
  - You can offer your customers a smoother, more streamlined PayPal Express Checkout experience, which tries alternative payment options when a customer's credit card is rejected
  - Improve the PayPal Express checkout experience by eliminating the following steps in the checkout process:
    - The order review page can be enabled or disabled
    - Eliminate the necessity of clicking **Update Order** before **Place Order**

(*Conversion* means helping customers stay interested and complete their purchases.)

## Security Enhancements

- Addressed a potential cross-site scripting (XSS) vulnerability while creating configurable product variants.
- Addressed a potential security issue that could result in displaying information about a

different order to a customer.

- Users can no longer change the currency if the payment method PayPal Website Payments Standard is used.
- Removed an .swf file from the Magento distribution because of security issues.
- Improved file system security.
- Enhanced the security of action URLs, such as billing agreements.
- Addressed a potential session fixation vulnerability during checkout.
- Improved the security of the Magento randomness function.

## Changes

- A default setting for configurable and bundled products has changed. When you create a configurable or bundled product in the Admin Panel, click **Manage > Products**. Create a new configurable or bundled product and click the **Design** tab. The default option for **Display Product Options In** has changed to **Product Info Column**.
- The Google Websites Optimizer has been disabled because it has been deprecated by Google. (In earlier Commerce versions, this option was available in the Admin Panel at **System > Configuration > SALES > Google API > Google Website Optimizer**).
- Two new options to prevent "[clickjacking](#)" if you run Magento in a `frame` or `iframe`:
  - Enable frames only in the same domain.
  - Enable frames.



**Note:** For security reasons, Magento strongly recommends against running the Magento software in a frame.

The options are available in the Admin Panel at **System > Configuration > ADVANCED > Admin > Security** and are named **Allow Magento Backend to run in frame** and **Allow Magento Frontend to run in frame**.

Enabling the option causes the [X-Frame-Options](#) request header to be sent.

- FedEx changed their endpoints for sandbox and production to the following:
  - Production: <https://ws.fedex.com/web-services>
  - Sandbox: <https://wsbeta.fedex.com/web-services>

Because of these changes, Magento cannot retrieve shipping rate information or print shipping labels for FedEx unless this fix is applied.

- Magento Commerce 1.14 includes a fix that prevented some Discover credit cards from validating properly. The issue was that certain Discover credit card number ranges were not recognized as being valid. As a result of the fix, all Discover cards should validate properly.



**Note:** This is *not* a security threat. No data has been compromised or misused. It affects only the ability to validate certain credit card number ranges as valid Discover card numbers.

- If you use the Solr search engine, you no longer need to manually reindex the Catalog Search Index (except after you upgrade to EE 1.14 from an earlier version).
- The default values for two configuration options changed. Both options can be found in

the Admin Panel under **System > Configuration > CATALOG > Catalog > Frontend**.

The new default values follow:

- **Products per Page on Grid Allowed Values** is now **12, 24, 36**.
- **Products per Page on Grid Default Value** is now **12**.
- On the New Rule page for shopping cart price rules (**Rule Information** tab page), explanatory text **Usage limit enforced for logged in customers only** was added to the **Uses per customer field**. This is to avoid confusion encountered by some Magento merchants.
- Changes to PayPal Express checkout:
  - Changes to both **Proceed to Checkout** and to **Pay With PayPal**:
    - Customers cannot edit their billing address on the PayPal site—in fact, the billing address does not display on the PayPal site.
    - Magento Commerce uses the customer's PayPal address information, not the address information stored in Magento Commerce. The customer does not need to enter Magento Open Source or Commerce address information.
    - When the customer is redirected to the PayPal site, they can click **Change** next to their payment method to change it.
  - The Magento administrator can set billing address information in the Admin Panel as follows: **System > Configuration > SALES > Payment Methods**. For any payment method that includes Express Checkout, in Basic Settings, set the value of the **Require Customer's Billing Address** list.
- Changes to **Pay With PayPal** only:
  - If the Magento administrator does not require the customer's billing address, the customer's billing address is set to the customer's PayPal *shipping* address.
  - If the Magento requires the customer's billing address, the customer's billing address is set to the customer's PayPal *billing* address.
  - On the PayPal site, the customer can click **Change** next to their shipping address to change it. The customer can choose any shipping address configured with PayPal.
  - Because the customer's configured address in Magento is not used, the customer's shipping address stored in PayPal is never changed.
- Changes to **Proceed to Checkout** only:
  - No **Change** link displays next to a customer's shipping address on the PayPal site.
  - In the event of a shipping address mismatch between Magento and PayPal, the following message displays when the customer attempts to pay using PayPal:

The address you entered on *store-name* is different than your PayPal preferred shipping address.  
Return to *store-name* if you'd like to change the shipping address

The user can select the **Use as preferred shipping address** checkbox to instruct PayPal to change their shipping address.

## Tax Calculation Fixes



Fixed price and dynamic price bundled products where the price is configured to include tax display prices correctly regardless of tax settings. (For example, customer's default tax rate is different from the origin tax rate.)

- Resolved a one-cent rounding issue when Fixed Product Tax (FPT) is enabled and the option **Apply Discount to FPT** is set to **Yes**. (These options are available in the Admin Panel by going to **System > Configuration > SALES > Tax > Fixed Product Taxes**.)
- Resolved issues with calculating the credit memo amount when FPT is discounted and the customer purchases more than one item.

## Fixes

Fixes in this release can be divided into the following categories:

- [Web Store and Shopping Cart Fixes](#)
- [Promotional Price Rule Fixes](#)
- [Administrative Ordering, Invoicing, Credit Memo Fixes](#)
- [Import Fixes](#)
- [Payment Method Fixes](#)
- [Solr Search Engine Fixes](#)
- [Other Fixes](#)

### Web Store and Shopping Cart Fixes

- With full page caching enabled:
  - Customers can no longer see each other's wish list or previously viewed products.
  - Related products set to not rotate display on the web store. (In other words, an administrator set **System > Configuration > CATALOG > Catalog > Rule-Based Product Relations**, option **Rotation Mode for Products in Related Product List** set to **Do not rotate**.)
  - The product ratings block refreshes on the storefront after a new rating has been approved.
  - Tier pricing is calculated properly.
- Resolved a 404 (Not Found) error navigating from a product in one category to an upsell product in a different category.
- A customer can update quantities of items in their mini shopping cart from their My Account page.
- The Minimum Advertised Price pop-up works properly in the web store. When the customer clicks **Click for price**, the price displays as expected.
- The "customer since" date is correct.
- Switching stores when viewing a product with store-scoped URL keys works as expected.
- Setting **System > Configuration > CATALOG > Inventory**, option **Display Out of Stock Products** to **Yes** no longer causes all products to appear as out of stock.

- Entering accented characters in the zip code field during checkout results in a validation error instead of an exception message.
- Gift card codes are sent only after an item is purchased.
- A customer who attempts to log in as another customer with incorrect credentials is denied.
- Resolved issues with applying a 100% discount to an order.
- Customers are no longer redirected to the home page when they have permission to view a category.
- Discount amount displays correctly for products with custom options.
- Issues with placing PayPal Payments Advanced or PayPal Payflow Link orders using Internet Explorer 9 have been resolved.

## Promotional Price Rule Fixes

The following fixes relate to administering and using shopping cart price rules and catalog price rules:

- Shopping cart price rules apply properly to grouped products.
- Two catalog price rules applied to the same product work properly.
- The setting **Stop Further Rules Processing** is honored.
- A user with read-only privileges in the Admin Panel cannot save changes to a price rule.
- Applying a shopping cart price rule does not display an exception.
- Coupon codes apply only to products eligible for the price rule.

## Administrative Ordering, Invoicing, Credit Memo Fixes

- An administrative user with access to only one website from which a product was deleted no longer sees a stack trace when attempting to create an RMA for that product. In other words, after a customer placed an order for a product on Website1, an administrator with privileges to all websites removes the product. Later, when an administrator with access to only Website1 attempts to create an RMA for the deleted product, that administrator no longer sees an error message; instead, they see an expected **Access Denied** message.
- Resolved an issue with incorrectly calculating the amount of an invoice when some items were discounted by a shopping cart price rule.
- Credit memo amount is calculated correctly when processing a partial invoice with a discount.
- Making comments in a credit memo no longer returns items to stock. (Prerequisite: an administrator set **System > Configuration > CATALOG > Inventory > Product Stock Options**, option **Automatically Return Credit Memo Item to Stock** set to **Yes**.)

## Import Fixes

- A product with a custom attribute set imports successfully.
- A fatal error caused by `Mage_ImportExport_Model_Scheduled_Operation` was

fixed.

Magento thanks Tim Bezhashvly for contributing to this fix.

## Payment Method Fixes

- If guest checkout is disabled, a customer must log in to check out with PayPal Express.
- Eliminated errors in the logs when an administrator clicks **System > Configuration > SALES > Payment Methods**.
- You can now use New Zealand dollars as the base currency with the eWAY Direct payment bridge.
- Store credit is applied correctly when using Website Payments Pro Hosted Solution.
- If the merchant country is Germany (DE), disabled guest checkout for the express checkout method and PayPal Website Payments Standard.

## Solr Search Engine Fixes

- Eliminated a spurious error from the logs when Solr is enabled as the search engine.
- Solr sorts search results by score.
- A quick search by SKU works with the Solr search engine if the SKU contains hyphen characters.
- Solr search works properly if you set `allow_url_fopen = Off` in `php.ini`.
- Using layered navigation filtering no longer returns the wrong results. (For example, filtering by brand works properly.)
- Resolved an issue where search results don't display correctly after a Magento upgrade.
- Catalog navigation works properly.
- Products display as expected in categories if the products have a Date attribute with the option **Used for Sorting in Product Listing** set to **Yes**. There are no exceptions in Magento logs after reindexing.
- Corrected the sort order of products searched by SKU.
- Search results of products with names and/or SKUs that contains numbers, letters, and a hyphen character (-) are as expected.
- Resolved issues with search results for products in a locale other than en\_US with numeric SKUs.
- Resolved issues with Solr not returning product search results.
- Search results no longer include products that are either Disabled or Out of Stock.

## Other Fixes

- Restored images on a customer's Reward Points page.
- Categories on the storefront now display with spaces between category names for cached and non-cached pages.
- A customer can now initiate a return from the storefront.

- An administrative user can subscribe to low stock RSS feeds without errors.
- Category URLs work as expected, regardless of the setting of **Create Custom Redirect for old URL** for the category's URL key.
- Setting `allow_url_fopen = Off` in `php.ini` has no effect on the CMS WYSIWYG editor.
- No fatal error displays when a role-restricted user previews a newsletter in the Admin Panel.
- Google Sitemap files now include the `.html` suffix for category and product URLs.
- Customers can use advanced search on your storefront if Magento EE is configured to use the default MySQL Fulltext search engine and the server uses MySQL 5.6.
- A role-restricted user can preview a newsletter in the Admin Panel to which the user has privileges.
- After synchronizing media files with the database, `media/customer/.htaccess` is present with the correct data. (Prerequisite: an administrator set **System > Configuration > ADVANCED > System > Storage Configuration for Media** set to **Database**).
- cron now restarts indexers if they previously failed to run.
- You can save changes to a category that has more than 1,000 products.
- Deactivating one of several banners no longer causes exceptions in `system.log`.
- Resolved issues with the WSDL cache.
- Improved the efficiency of product searches.
- Resolved issues with the full page cache crawler.
- Resolved issues with the DHL International shipping method.
- Resolved 404 (Not Found) errors in layered navigation.
- Resolved a SQL error when attempting to assign a bundled product to another website.
- Rules-based product relations perform as expected after being saved.
- Resolved an issue with sending duplicate **Content-Type** headers when using `mod_fastcgi` with the Apache web server.

## Open Source Software Licensing Agreements

Some versions of Magento Commerce use open source software licensing. Following are license agreements for that software.

**Touch punch:** This code is dual licensed under the MIT or GPL Version 2 licenses and is therefore free to use, modify and/or distribute, but if you include Touch Punch in other software packages or plugins, please include an attribution to the original software and a link to this [Touch Punch website](#).

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Responsive Email Developer's Guide for Magento EE 1.14.1 and Magento CE 1.9.1

by Erik Hansen

[Edit this page on GitHub](#) 

## Table of Contents

- [Introduction](#)
- [Using cron to Send Emails](#)
- [Supported Email Clients and Devices](#)
- [Important Information about Upgrading](#)
- [Customizing the Header or Footer in Transactional Emails](#)
- [Customizing CSS Styles for Emails](#)
- [Using a High-Resolution Logo](#)
- [Using Contact Information in Transaction Emails](#)
- [Localizing Transactional Emails](#)
- [Using Updated Newsletter Templates](#)
- [Getting Help](#)

## Introduction

We improved transactional emails and newsletter templates in Magento Enterprise 1.14.1 and Magento Community 1.9.1 in a number of ways, including:

- Emails are now optimized for any screen size, including mobile and tablet devices
- Header and footer content for transactional emails are now contained in their own files
- CSS is now contained in separate CSS files
- We provide you with a responsive sample newsletter template



**Note:** This article focuses on responsive emails but the same techniques can be used for newsletter templates as well.  
For more information about newsletter templates, see [Using Updated Newsletter Templates](#).

## Using cron to Send Emails

Starting in Magento EE 1.14.1 and CE 1.9.1, your Magento cron job sends *all* emails, including transactional emails. You must configure cron for emails to work.

For more information, see [Setting Up Cron Jobs](#).

## Supported Email Clients and Devices

We tested responsive emails using a combination of real devices and [Litmus](#). Due to the greatly varied level of support among email clients for modern web technologies, not all email clients rendered the emails perfectly. However, all of the following clients should render the emails in a manner that allows them to be easily read without obvious issues. Refer to the following sections for more information.

### Supported Desktop Clients

| OPERATING SYSTEM                                                                | CLIENTS SUPPORTED                                                                                                                   |
|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| OS X 10.8                                                                       | Apple Mail 6                                                                                                                        |
| OS X 10.7                                                                       | Outlook 2011                                                                                                                        |
| <ul style="list-style-type: none"><li>• Windows 8</li><li>• Windows 7</li></ul> | <ul style="list-style-type: none"><li>• Outlook 2013</li><li>• Outlook 2010</li><li>• Outlook 2007</li><li>• Outlook 2003</li></ul> |

### Supported Mobile Clients

| OPERATING SYSTEM            | CLIENTS SUPPORTED                                                                                            |
|-----------------------------|--------------------------------------------------------------------------------------------------------------|
| Android 4.2 ("Jelly Bean")  | Native email app                                                                                             |
| Android 2.3 ("Gingerbread") | Native email app                                                                                             |
| Gmail App (Android 4.2)     | Native email app                                                                                             |
| Blackberry 5 OS             | Native email app                                                                                             |
| iOS 8                       | <ul style="list-style-type: none"><li>• iPhone 6</li><li>• iPhone 6 Plus</li></ul>                           |
| iOS 7                       | <ul style="list-style-type: none"><li>• iPad (Retina)</li><li>• iPad Mini</li><li>• iPhone 5s</li></ul>      |
| iOS 6                       | <b>Mail on these devices:</b> <ul style="list-style-type: none"><li>• iPhone 5</li><li>• iPhone 4s</li></ul> |

### Supported Web Clients

| EMAIL APPLICATION | BROWSERS SUPPORTED*                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------|
| AOL Mail          | <ul style="list-style-type: none"><li>• Chrome</li><li>• Internet Explorer</li><li>• Firefox</li></ul> |
| Gmail             | <ul style="list-style-type: none"><li>• Chrome</li><li>• Internet Explorer</li><li>•</li></ul>         |

|             |                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------|
|             | Firefox                                                                                                    |
| Yahoo! Mail | <ul style="list-style-type: none"> <li>• Chrome</li> <li>• Internet Explorer</li> <li>• Firefox</li> </ul> |
| Outlook.com | <ul style="list-style-type: none"> <li>• Chrome</li> <li>• Internet Explorer</li> </ul>                    |

<sup>1</sup>The latest version of each browser was used for testing.

## Important Information about Upgrading

This section applies to anyone who used transactional emails in a version earlier than EE 1.14.1 or CE 1.9.1.

After upgrading from an earlier Magento version, you should preview your transactional emails, looking for changes in the logo, fonts, and styles. Because of changes introduced in this release, you might notice some differences.

Also, in CE 1.9.1 and EE 1.14.1 for the first time, you must specify a height and width for your logo image; the dimensions must be set to the area in which the logo displays. For more information, see [Using a High-Resolution Logo](#).

## Customizing the Header or Footer in Transactional Emails

In earlier Magento CE and EE versions, each transactional email template contained a copy of the HTML for the header and footer. To make it easier to customize transactional emails, the header and footer HTML is now centralized in the following template files:

- [your Magento install dir]/app/locale/en\_US/template/email/html/header.html
- [your Magento install dir]/app/locale/en\_US/template/email/html/footer.html

The header and footer template files get included in the transactional emails using a new `template` directive. For an example of how this works, open `app/locale/en_US/template/email/account_new.html`. It contains two directives that include `header.html` and `footer.html`:

```
{{template config_path="design/email/header"}}
```

```
{{template config_path="design/email/footer"}}
```

You can customize the header and footer as discussed in the following sections:

- [Customizing Header and Footer Templates Using Files](#)
- [Customizing Header and Footer Templates Using the Magento Admin Panel](#)

## Customizing Header and Footer Templates Using Files

If you can access the Magento server file system, you can customize the header and footer by editing the following files:

- [your Magento install dir]/app/locale/en\_US/template/email/html/header.html
- [your Magento install dir]/app/locale/en\_US/template/email/html/footer.html



**Note:** To make changes to the templates at the website or store view configuration scopes, see the next section.

## Customizing Header and Footer Templates Using the Magento Admin Panel

You must use the Magento Admin Panel to customize header and footer templates if any of the following are true:

- You cannot access the Magento server file system
- To customize the templates for a unique website, store, or store view

The example discussed in the following sections shows how to customize the header template. You can use the same procedure to customize the footer as well.

Using custom templates is a two-step process: first, define the template, and second, specify the website or store view on which it's used. See one of the following sections for more information:

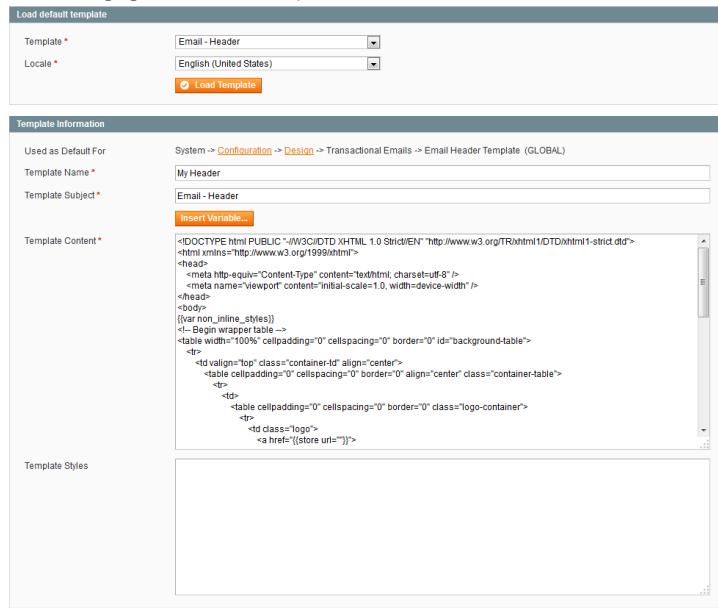
- [Customizing the Template in the Admin Panel](#)
- [Using the Template in a Website or Store View](#)

## Customizing the Template in the Admin Panel

1. Log in to the Magento Admin Panel as an administrator.
2. Click **System > Transactional Emails**.
3. On the Transactional Emails page, click **Add New Template**.
4. From the **Template** list, click **Email - Header**.
5. From the **Locale** list, click the template's locale.
6. Click **Load Template**.
7. Enter the following information:

| ITEM                          | DESCRIPTION                                                                                                                    |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>Template Name</b> field    | Enter a name for the new template, such as My Header                                                                           |
| <b>Template Subject</b> field | Ignore; this field isn't used.                                                                                                 |
| <b>Insert Variable</b> Button | Click to see a list of variables to insert. For more information about variables, see the <a href="#">Magento User Guide</a> . |
| <b>Template Content</b> field | Enter the contents of your email template.                                                                                     |
| <b>Template Styles</b> field  | Ignore; this field is not supported when customizing the header and footer templates.                                          |

The following figure shows an example:



8. Click **Preview Template** to look at the HTML code in a separate browser tab page or window, or click **Save Template** to save it.

## Using the Template in a Website or Store View

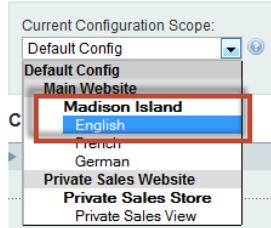
This section discusses how to use the email template you just created in the global configuration, a website, or a store view.

To use this template:

1. In the Admin Panel, click **System > Configuration > GENERAL > Design**.
2. From the **Current Configuration Scope** in the upper left corner, click the scope.

Initially, the template is associated with **Default Config**, which means it applies to all websites and store views. You can optionally associate it with a specific website or store view by selecting it from the list.

The following example shows how to associate a template with the Madison Island English store view.



3. In the right pane, click **Transactional Emails** to expand it.
4. From the **Email Header Template** list, click the header template you created earlier.

The following figure shows an example:

5. Click **Save Config** in the upper right corner of the page.

Now transactional emails use the header or footer template you created.

## Customizing CSS Styles for Emails

CSS styles for transactional emails are "inlined" on each HTML tag because many email clients require it.

CE 1.9.1 and EE 1.14.1 simplifies the way you customize email templates by locating all CSS styles in separate CSS files. When transactional emails are sent, the CSS styles are loaded from the separate CSS files and are "inlined" in the HTML before being sent.

Inlining is done using the [Emogrifier](#) library, which takes the HTML and CSS and adds all of the CSS styles to `<style>` attributes on the HTML tags.

The CSS for the emails is contained in the following Sass files:

- [your Magento install dir]/skin/frontend/rwd/default/scss/email-inline.scss
- [your Magento install dir]/skin/frontend/rwd/default/scss/email-non-inline.scss

As you might expect, these two files compile to `skin/frontend/rwd/default/css/email-inline.css` and `skin/frontend/rwd/default/css/email-non-inline.css`, respectively. If you're not familiar with how to work with Sass files, refer to the [RWD theme documentation](#).

## Using Inline Styles to Customize Email CSS

The `email-inline.css` file contains all of the styles that the Emogrifier class applies to the HTML before an email is sent. Magento applies only these styles to an email template if it contains this new directive:

```
{{inlinecss file="email-inline.css"}}
```

Notice that the templates in `app/locale/en_US/template/email/` include this directive in each email template. The directive enables you to specify CSS that loads from the current storefront's theme and is inlined on that template.

The directive supports the standard Magento fallback hierarchy, which means that if the `email-inline.css` can't be found in the store's theme, Magento looks in each of the parent themes until a match is found. (A sample theme directory is

`skin/frontend/custompackage/customtheme/css/email-inline.css)`

Refer to the [Emogrifier README](#) to see what CSS selectors are supported.



**Note:** The `{{{inlinecss file=""}}}` directive can be used a single time and cannot be used in templates that get included by other templates (for example, `app/locale/en_US/template/email/html/header.html` and `app/locale/en_US/template/email/html/footer.html`).

## Using Non-Inline Styles to Customize Email CSS

There are certain CSS styles that can't be applied inline, such as media queries styles. These styles must be in a `<style type="text/css"></style>` tag.

To understand how non-inline styles are included, see line 7 in `app/locale/en_US/template/email/html/header.html`:

The variable `{{var non_inline_styles}}` loads CSS from several sources:

- The contents of `email-non-inline.css`.

The file loads from the current storefront's theme and supports the same fallback structure as the `email-inline.css` file. For Magento to load additional files, go to **System > Configuration > GENERAL > Design**, click the **Transactional Emails** section and change the value of the **Non-inline CSS File(s)** field.

- Each of the transactional emails has a comment at the top of the template in this format:

```
<!--@styles
@-->
```

If you modify a template and add CSS styles inside that comment, they are included in the `non_inline_styles` variable.

- If you customize transactional emails using the Magento Admin Panel, you can add CSS styles to the **Template Styles** field to include those styles in the `non_inline_styles` variable. For more information, see [Customizing Header and Footer Templates Using the Magento Admin Panel](#).

After the CSS is loaded, it's wrapped in a `<style type="text/css"></style>` tag and is assigned to the `non_inline_styles` variable.



**Note:** When you are writing "non-inline" CSS, you must add the `!important` declaration after each property so the style has a high enough specificity to get applied. This is the best way to get your custom styles to override styles defined in the `email-inline.css` file.

## Using a High-Resolution Logo

For the logo in your email to look optimal on high-resolution devices such as retina displays, you must use a logo that is at least two times larger than the area in which you want it to display. For example, let's say your email has a  $200\text{px} \times 100\text{px}$  area for the logo. The logo image must be at least  $400\text{px} \times 200\text{px}$ .

If you don't have access to a high-resolution version of your logo, you can upload a normal-resolution image. For example, if your logo image is  $200\text{px} \times 100\text{px}$ , specify **200** for the width and **100** for the height.

## Upgrade Considerations

This section applies to you if you upgraded to CE 1.9.1 or EE 1.14.1 from an earlier version.

In CE 1.19.1 and EE 1.14.1 for the first time, transactional email templates specify the width and height of the area in which the logo displays. Therefore, if you use transactional emails, you must configure your templates with a logo height and width as discussed in the next section.

For example, suppose the area in which the logo displays is  $200\text{px} \times 100\text{px}$ . Specify **200** for height and **100** for width even if the physical dimensions of the logo are larger.

## Adding a High-Resolution Logo

You can add your logo to transactional emails in either of the following ways:

- [Updating a High-Resolution Logo Using the File System](#)
- [Updating a High-Resolution Logo Using the Admin Panel](#)

## Updating a High-Resolution Logo Using the File System

If you can access the Magento server file system, follow these steps:

1. Locate the skin directory for your store (for example, `skin/frontend/custompackage/customtheme/`).
2. Save the logo image as `logo_email.gif` in the skin directory (for example, `skin/frontend/custompackage/customtheme/images/logo_email.gif`).
3. Open `app/locale/en_US/template/email/html/header.html` in a text editor.

4. Update the **width** and **height** attributes to reflect the area in which you want your logo to display (for example, 200px x 100px).  
A snippet follows:

```

```



**Note:** To use a file type other than GIF or if you need to upload unique logos for different websites, stores, or store views, see the next section.

## Updating a High-Resolution Logo Using the Admin Panel

You can update your high-resolution logo using the Admin Panel if any of the following are true:

- The file format is not GIF
- You don't have access to the Magento server file system
- The logo is specific to a website, store, or store view

To upload a high-resolution logo using the Admin Panel:

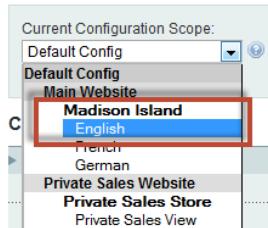
1. Log in to the Magento Admin Panel as an administrator.

2. Click **System > Configuration > GENERAL > Design**.

3. From the **Current Configuration Scope** in the upper left corner, click the scope to display the logo.

Initially, the logo is associated with **Default Config**, which means it applies to all websites and store views. You can optionally associate it with a specific website or store view by selecting it from the list.

The following example shows how to associate a logo with the Madison Island English store view.



4. In the right pane, click **Transactional Emails** to expand it.

5. Enter the following information:

| ITEM                         | DESCRIPTION                                                                                                                    |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>Logo Image</b>            | Click <b>Browse</b> and browse to locate the logo image.                                                                       |
| <b>Delete Image</b> checkbox | Select the checkbox to delete the current logo image. Ignore the other fields.                                                 |
| <b>Logo Image Alt</b> field  | Enter alternate text to display if the image is not available. For a logo, it's typically the name of your company.            |
| <b>Logo Width</b> field      | Enter the numeric width, in pixels, of the area in which to display the image ( <i>not</i> the dimensions of the logo image).  |
| <b>Logo Height</b> field     | Enter the numeric height, in pixels, of the area in which to display the image ( <i>not</i> the dimensions of the logo image). |

The following figure shows an example.

- Click **Save Config** in the upper right corner of the page.

## Using Contact Information in Transaction Emails

Transactional emails can output your store email address, store phone number, and store hours of operation if those values are configured in the Admin Panel. To set those values:

- Log in to the Magento Admin Panel as an administrator.
- To set the phone number and hours of operation:
  - Click **System > Configuration > GENERAL > General**.
  - In the right pane, click **Store Information** to expand it.
  - If you want to include a phone number, input the phone number in the **Store Contact Telephone** field.
  - If you want to include your hours of operation, input them in the **Store Hours of Operation** field.
- To set the store email address:
  - Click **System > Configuration > GENERAL > Store Email Addresses**.
  - In the right pane, click **General Contact** or **Sales Representative** to expand it.
  - In the **Sender Name** field, input the name that you want the email to be sent from, such as your company name.
  - In the **Sender Email** field, input the email address.
- When you're finished, click **Save Config** in the upper right corner of the page.

## Localizing Transactional Emails

Only the en\_US templates were updated to the new responsive email structure. To use the responsive emails for other locales, copy the en\_US/template/email/ directory into the respective app/locale/[LOCALE\_CODE] directory and update the text strings to that locale.

## Using Updated Newsletter Templates

The focus of this article is on transactional emails but the same techniques can be used with newsletter templates as well, including:

- Import the header and footer using the `{template config_path=""}` directives
- Apply inline styles using the `{inlinecss file=""}` directive
- Include non-inline styles using `{var non_inline_styles}`

In the Magento Admin Panel, go to **Newsletter > Newsletter Templates** and look at the template named **Example Newsletter Template**. We added this template as a reference for how to build a responsive newsletter.

## Getting Help

If you encounter issues customizing email templates, contact Magento Support. Another option is to post your question on the community

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento Community Edition (CE) 1.9 and Magento Enterprise Edition (EE) 1.14 Responsive Web Design Developer's Guide

by Erik Hansen

[Edit this page on GitHub](#) 

## Contents

- [Introduction](#)
- [Supported Devices and Browsers](#)
- [Understanding the Changes in the Responsive Theme](#)
- [Installing Compass and Sass](#)
- [Disabling the Magento Cache](#)
- [Copying Files to Create Your Custom Theme](#)
- [Responsive Theme Reference](#)
- [Removing Sample Data Assets](#)
- [Advanced Topics](#)
- [Getting Help With Your Theme](#)

## Introduction

Magento Enterprise Edition 1.14 and Magento Community Edition 1.9 come with a new theme named `rwd` that implements [Responsive Web Design \(RWD\) practices](#). This guide discusses what you need to know when customizing your own responsive theme.

Magento CE's and EE's responsive theme uses a number of new technologies:

- [Sass/Compass](#) : A CSS pre-compiler that provides organized, reusable CSS.
- [jQuery](#) : Used for all custom JavaScript in the responsive theme. jQuery operates in `noConflict()` mode so it doesn't conflict with Magento's existing JavaScript library.
- Additional JavaScript libraries discussed in more detail in [Working With JavaScript](#).

If you're not familiar with Sass or Compass, you should familiarize yourself with them before continuing with this guide.

- [Sass](#)
  - [Quick-Start Guide](#)
  - [Full Documentation](#)

- [Compass](#)

The new responsive theme code is contained in the following directories (relative to your Magento installation directory):

- app/design/frontend/rwd
- skin/frontend/rwd

See one of the following sections for more information:

- [Intended Audience](#)
- [Documentation Assumptions and Conventions](#)
- [Exceptions to Responsive Web Design](#)

## Intended Audience

This guide assumes that you have a good understanding of Magento frontend development and Responsive Web Design techniques.

## Documentation Conventions and Assumptions

This section discusses some basic information that applies to this guide.

### Documentation Conventions

This guide uses the names `custompackage` and `customtheme` for the design package and theme. When you build your own responsive theme, use a more descriptive name, such as the name of your company.

The variable value `[your Magento install dir]` refers to the directory in which you installed Magento CE or EE. Typical values follow:

- Ubuntu: `/var/www/magento`
- CentOS: `/var/www/html/magento`

### Assumption: How You Customize CSS

You have two choices to write your custom CSS styles:

- Write CSS in the Sass stylesheets using the same structure that the Magento responsive theme uses. This approach requires setting up a Compass compiler, but is *strongly* recommended. This guide assumes you'll use this approach.
- Directly modify the CSS files that ship with the responsive theme and ignore the Sass files. While this approach might be simpler in the short run, the CSS files in the responsive theme were output by the Sass compiler and are therefore not conducive to being customized directly. This approach is *not* recommended.

## Exceptions to Responsive Web Design

Responsive design was *not* implemented for the following:

- Multiple address checkout
- Popular search terms page
- Popular search terms cloud
- Site map
- My Account—Billing Agreements
- My Account—Recurring Profiles
- E-mail templates
- Tags
- Polls
- Captcha
- The gift registry (*EE only*)
- Multiple wishlists (*EE only*)
- Place order by SKU (*EE only*)



**Note:** Magento still supports other non-responsive themes as well. You don't need to use this guide to customize those themes. For more information, see the [Designer's Guide to Magento](#)

## Supported Devices and Browsers

Magento's responsive theme was developed with full support for the following devices and browsers:

- Windows:
  - Internet Explorer 9 and later
  - Latest stable version of Chrome
  - Latest stable version of Firefox
- OS X:
  - Latest stable version of Chrome
  - Latest stable version of Firefox
  - Latest stable version of Safari
- iOS (iPhone and iPad): Safari on iOS 7
- Android: Chrome browser on Android 4.0+ (also known as Ice Cream Sandwich)

Basic compatibility should be expected on Android 2.3+ (and other mobile devices) and Internet Explorer 8, but full support is not guaranteed.

## Understanding the Changes in the Responsive Theme

The new responsive theme in EE 1.14 and CE 1.9 is based on the existing base/default theme. The new responsive theme has about 130 template files that have been modified to better handle the needs of responsive development. Following are some of the changes that have been made in these template files:

- Rearranged elements to work better with smaller viewports.
- Changed the Doctype to html to enable use of HTML5 elements. (To avoid large amounts of template modifications, HTML5 tags were implemented only when necessary.)
- Refactored attributes of <input> elements to HTML5 standards.
- Refactored product lists to show varying number of products per column.
- Removed all <fieldset> elements because they don't work well in a responsive context.

## Changes to Layout Files

A number of the core layout files (`catalog.xml`, `customer.xml`) were slightly modified to account for responsive changes. There are only a few notable changes to the layout files:

- `[your Magento install dir]/app/design/frontend/rwd/default/layout/page.xml` was modified to include all of the new responsive assets.
- Because two and three column layouts collapse to a single column on smaller viewports, all content in the `left` block are displayed under the main content. This is a problem because blocks—like the navigation on all My Account pages—must display above the main content on smaller viewports. To address this, a new `left_first` block was added that looks like the left column on larger viewports, but on smaller viewports, it displays above the main content.

## Using the New Theme/Layout Structure—***Magento EE only***

If you've worked with a previous version of Magento EE, you're aware that the enterprise/default theme contains copies of *many* of the layout and template files from base/default—some with very minor changes. This is not an optimal approach because as it results in duplicated content.

The Magento EE responsive theme's `rwd/enterprise` directory includes only template and layout files that contain Magento EE-only features.

Instead of copying layout files as the enterprise/default theme does, `rwd/enterprise` isolates changes in their own layout files in `[your Magento install dir]app/design/frontend/rwd/enterprise/layout/enterprise`

The `rwd/enterprise` theme replaces the `enterprise/default` theme and contains *far* fewer files than its predecessor (156 compared to 280). This new approach makes it easier for you to see what functionality is unique to Magento EE.

## Changes to Magento Theme Fallback

Magento CE 1.9 and EE 1.14 both introduce a theme configuration file named `theme.xml`. This file allows theme developers to specify a parent theme which results in a more predictable and useful fallback than earlier versions.

For more information, see [Alan Storm's Parent/Child Theme article](#).

## Installing Compass and Sass

Sass is a powerful CSS pre-compiler that extends CSS supports variables, mixins, and placeholders. Compass is a library for Sass that adds hundreds of useful mixins and functions to Sass—everything from color blending to sprite generation.

Because the new Magento responsive theme uses the Compass library to compile its Sass files into CSS, you must install Compass in your

development environment in any of the following ways:

- [Command line](#)
- Using any of the following third-party applications:
  - [Scout](#)
  - [Compass App](#)
  - [CodeKit](#)
- Compile using your code editor (PhpStorm, Sublime, and so on): Many code editors run Compass from the editor so that you don't have to use the command line to re-compile when you make changes. However, you must install Compass using the command line.

## Disabling the Magento Cache

A good practice during development on Magento is to disable Magento's cache. Although doing this adversely impacts performance, it frees you from having to manually clear the cache when you make changes to files or configuration settings that are being cached.

You should always enable the cache in a production environment.

To disable the cache:

1. Log in to the Magento Admin Panel as an administrator.

2. Click **System > Configuration > Cache Management**.

3. On the Cache Management page, click **Select All**.

**Note:** The **Page Cache** option at the bottom of the list is available only in Magento EE.

4. From the **Actions** list in the upper right corner of the page, click **Disable**.

5. Click **Submit**.

The following figure shows an example.

| Cache Storage Management                                       |                                                                            |                                                                                                                             |
|----------------------------------------------------------------|----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
|                                                                |                                                                            | Actions   Disable   Submit                                                                                                  |
| <input type="checkbox"/> Select All                            | <input type="checkbox"/> Unselect All                                      | <input type="checkbox"/> Select Visible                                                                                     |
| <input checked="" type="checkbox"/> Configuration              | Description                                                                | Associated Tags                                                                                                             |
| <input checked="" type="checkbox"/> Layouts                    | System(config.xml, local.xml) and modules configuration files(config.xml). | CONFIG<br>LAYOUT_GENERAL_CACHE_TAG<br>BLOCK_HTML<br>TRANSLATE<br>COLLECTION_DATA<br>EAV<br>CONFIG_API<br>CONFIG_API2<br>FPC |
| <input checked="" type="checkbox"/> Blocks HTML output         | Page blocks HTML.                                                          | ENABLED<br>ENABLED<br>ENABLED<br>ENABLED<br>ENABLED<br>ENABLED<br>ENABLED<br>ENABLED<br>ENABLED                             |
| <input checked="" type="checkbox"/> Translations               | Translation files.                                                         |                                                                                                                             |
| <input checked="" type="checkbox"/> Collections Data           | Collection data files.                                                     |                                                                                                                             |
| <input checked="" type="checkbox"/> EAV types and attributes   | Entity types declaration cache.                                            |                                                                                                                             |
| <input checked="" type="checkbox"/> Web Services Configuration | Web Services definition files (api.xml).                                   |                                                                                                                             |
| <input checked="" type="checkbox"/> Web Services Configuration | Web Services definition files (api2.xml).                                  |                                                                                                                             |
| <input checked="" type="checkbox"/> Page Cache                 | Full page caching.                                                         |                                                                                                                             |

6. Click **Submit**.

The following figures shows all cache types disabled.

| Cache Storage Management                                       |                                                                            |                                                                                                                             |
|----------------------------------------------------------------|----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
|                                                                |                                                                            | Actions   Refresh   Submit                                                                                                  |
| <input type="checkbox"/> Select All                            | <input type="checkbox"/> Unselect All                                      | <input type="checkbox"/> Select Visible                                                                                     |
| <input checked="" type="checkbox"/> Configuration              | Description                                                                | Associated Tags                                                                                                             |
| <input checked="" type="checkbox"/> Layouts                    | System(config.xml, local.xml) and modules configuration files(config.xml). | CONFIG<br>LAYOUT_GENERAL_CACHE_TAG<br>BLOCK_HTML<br>TRANSLATE<br>COLLECTION_DATA<br>EAV<br>CONFIG_API<br>CONFIG_API2<br>FPC |
| <input checked="" type="checkbox"/> Blocks HTML output         | Page blocks HTML.                                                          | DISABLED<br>DISABLED<br>DISABLED<br>DISABLED<br>DISABLED<br>DISABLED<br>DISABLED<br>DISABLED<br>DISABLED                    |
| <input checked="" type="checkbox"/> Translations               | Translation files.                                                         |                                                                                                                             |
| <input checked="" type="checkbox"/> Collections Data           | Collection data files.                                                     |                                                                                                                             |
| <input checked="" type="checkbox"/> EAV types and attributes   | Entity types declaration cache.                                            |                                                                                                                             |
| <input checked="" type="checkbox"/> Web Services Configuration | Web Services definition files (api.xml).                                   |                                                                                                                             |
| <input checked="" type="checkbox"/> Web Services Configuration | Web Services definition files (api2.xml).                                  |                                                                                                                             |
| <input checked="" type="checkbox"/> Page Cache                 | Full page caching.                                                         |                                                                                                                             |

## Copying Files to Create Your Custom Theme

This section discusses how to start building a custom responsive theme.

See one of the following sections for more information:

- [Step 1: Creating a Directory Structure for Your Responsive Theme](#)
- [Step 2: Copying Files For Your Theme](#)
- [Step 3: Configuring a Compass Watcher](#)
- [Step 4: Configuring Magento to Load Your Theme](#)
- [Step 5: Verifying Your Custom Theme](#)
- [Troubleshooting Theme Configuration](#)

## Step 1: Creating a Directory Structure for Your Responsive Theme

You'll do your responsive theme development using a copy of the responsive theme provided with Magento CE and EE. This section discusses how to create new top-level directories for your theme and to copy files to those directories.

The `rwd/default` theme falls back directly to the `base/default` theme, so themes in the `default` package will never be included as a part of the fallback chain.



**Note:** *Magento EE only.* The `enterprise/default` theme is completely replaced by `rwd/enterprise` and therefore, `enterprise/default` is not included as a part of the fallback chain.

The top-level directories you'll need are:

- [your Magento install dir]/app/design/frontend/custompackage/customtheme
- [your Magento install dir]/skin/frontend/custompackage/customtheme

To create those directories and to add your theme configuration:

1. Using a command shell, log in to your Magento host as a user with privileges to write to the web server docroot. (Typically, this is the web server user.)

2. Enter the following commands in the order shown:

```
cd [your Magento install dir]
mkdir -p app/design/frontend/custompackage/customtheme
mkdir -p skin/frontend/custompackage/customtheme
```

3. Create another directory for your theme configuration and change to that directory:

```
mkdir app/design/frontend/custompackage/customtheme/etc
cd app/design/frontend/custompackage/customtheme/etc
```

4. Use vim or another text editor to create your theme configuration file named `theme.xml` there:

```
vim theme.xml
```

5. `theme.xml` must have different contents for Magento CE or Magento EE:

- Magento CE:

```
<?xml version="1.0"?>
<theme>
 <parent>rwd/default</parent>
</theme>
```

- Magento EE:

```
<?xml version="1.0"?>
<theme>
 <parent>rwd/enterprise</parent>
</theme>
```



**Important:** It's very important you have a `theme.xml` in the `app/design/frontend/custompackage/customtheme/etc` directory with exactly the contents shown. Failure to configure `theme.xml` correctly prevents Magento from loading your theme.

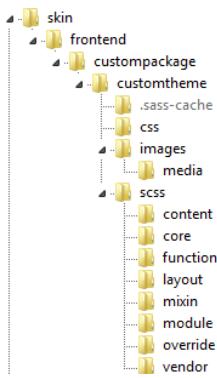
## Step 2: Copying Files For Your Theme

Use the following commands to copy files for your custom theme:

```
cd [your Magento install dir]
cp -R skin/frontend/rwd/default/images skin/frontend/custompackage/customtheme
cp -R skin/frontend/rwd/default/scss skin/frontend/custompackage/customtheme
```

Copying the `images` directory is important because the CSS uses relative image paths. To avoid duplicating Sass files, review the information discussed in [Sass Fallback Structure](#).

The following figure shows how your custom theme directory structure looks after copying these files and directories.



## Step 3: Configuring a Compass Watcher

You can configure Compass to watch your skin directories for changes to CSS and JavaScript and to compile them when something changes.

You can do this in either of the following ways:

- (*Recommended*) Using your PHP IDE. For example, if you use PhpStorm, see [Working with Sass and SCSS in Compass Projects](#) and [Using File Watchers](#).
- Using following commands:

```
cd [your Magento install dir]/custom-theme-dir/scss
compass watch .
```

For example, if Magento is installed in /var/www/html/magento, enter:

```
cd /var/www/html/magento/skin/frontend/custompackage/customtheme/scss
compass watch .
```

For more information, see [Compass Command Line Documentation](#).

## Success Messages

Messages similar to the following display to indicate the command completed successfully:

```
>>> Change detected at 09:57:12 to: madisonisland-ie8.scss
create ../../css/madisonisland-ie8.css
create ../../css/styles.css
create ../../css/madisonisland.css
create ../../css/styles-ie8.css
create ../../css/scaffold-forms.css
>>> Compass is polling for changes. Press Ctrl-C to Stop.
```

## Failure Message

If the following message displays, it likely indicates that config.rb is not in the correct location:

```
Nothing to compile. If you're trying to start a new project, you have left off the directory argument.
```

Enter the following commands to verify you have config.rb in the correct location:

```
cd [your Magento install dir]
ls skin/frontend/custompackage/customtheme/scss
```

If there is no config.rb in that location, enter the following command:

```
cp skin/frontend/rwd/default/scss/config.rb skin/frontend/custompackage/customtheme/scss
```

Compass requires a config.rb file located in the directory from which you run this command. The sample provided with Magento CE and EE follows:

```
note: this should never truly be referenced since we are using relative assets
http_path = "/skin/frontend/rwd/default/"
css_dir = "../css"
sass_dir = "../scss"
images_dir = "../images"
javascripts_dir = "../js"
relative_assets = true
```

```
output_style = :expanded
environment = :production
```

After correcting the issue, try the `compass watch` commands discussed in [Step 3: Configuring a Compass Watcher](#).

## Step 4: Configuring Magento to Load Your Theme

This section discusses how to configure Magento to load your new theme.

1. Log in to the Magento Admin Panel as an administrator.
2. Click **System > Configuration > GENERAL > Design**.
3. In the right pane, click **Package** to expand it.
4. In the **Current Package Name** field, enter `custompackage`
5. In the right pane, expand **Themes**.
6. In the **Default** field, enter `customtheme`

The following figure shows an example.

The screenshot shows the 'Design' configuration page in the Magento Admin Panel. The 'Package' section is expanded, showing the 'Current Package Name' set to 'custompackage'. The 'Themes' section is expanded, showing the 'Default' theme set to 'customtheme'. Both sections have an 'Add Exception' button and a note about matching expressions in the same order as displayed in the configuration.

7. In the top right corner of the page, click **Save Config**.  
If an error displays when you attempt to save the configuration, see [Troubleshooting Theme Configuration](#).
8. If prompted, flush the cache:
  - a. Click **System > Configuration > Cache Management**.
  - b. At the top of the page, click **Flush Magento Cache**.

## Step 5: Verifying Your Custom Theme

To verify that Magento loads your theme:

1. Navigate to your web store in a web browser.
2. View the web page source.
3. If you see a reference to `/skin/frontend/custompackage/customtheme/css/styles.css`, you are ready to begin building your new theme.



**Note:** If your theme fails to load properly, see the next section.

## Troubleshooting Theme Configuration

See one of the following sections for troubleshooting assistance:

- [General Troubleshooting](#)
- [Directory and File Reference](#)

### General Troubleshooting

The following sections assist you in determining the cause of errors you might experience after setting up your custom theme.

## No formatting on the web store

*Description:* Most likely an issue with `theme.xml`. You can confirm this by viewing the page source and searching for `styles.css`. If `styles.css` is loading from a path similar to the following, `theme.xml` is most likely not loading.

```
http://www.example.com/magento/skin/frontend/base/default/css/styles.css
```

*Solution:* Enter the following command:

```
cat [your Magento install dir]/app/design/frontend/custompackage/customtheme/etc/theme.xml
```

If the error `No such file or directory` displays, check your directory paths and create `theme.xml` as discussed in [Step 2: Copying Files For Your Theme](#).

If you have a `theme.xml`, check to make sure it's configured *exactly* as shown [here](#).

## Error saving the custom package and theme in the Admin Panel

*Symptom:* The following error displays:

```
An error occurred while saving this configuration: package with this name does not exist and cannot be set.
```

*Description:* The package and theme names you entered in the Admin Panel must be directories in a specific location on the file system. These directories don't exist. (See the figure in [Directory and File Reference](#).) Check the path and spelling.

*Solution:* Repeat the tasks discussed in [Step 1: Creating a Directory Structure for Your Responsive Theme](#).

## Wrong formatting in the web store

*Symptom:* Examples:

- No images display
- Only a list of categories displays on the left side of the page, with images after the text

*Description:* There is a different `theme.xml` for CE and EE. You might have used an EE `theme.xml` with CE or vice versa.

You can verify this using a web browser inspector:

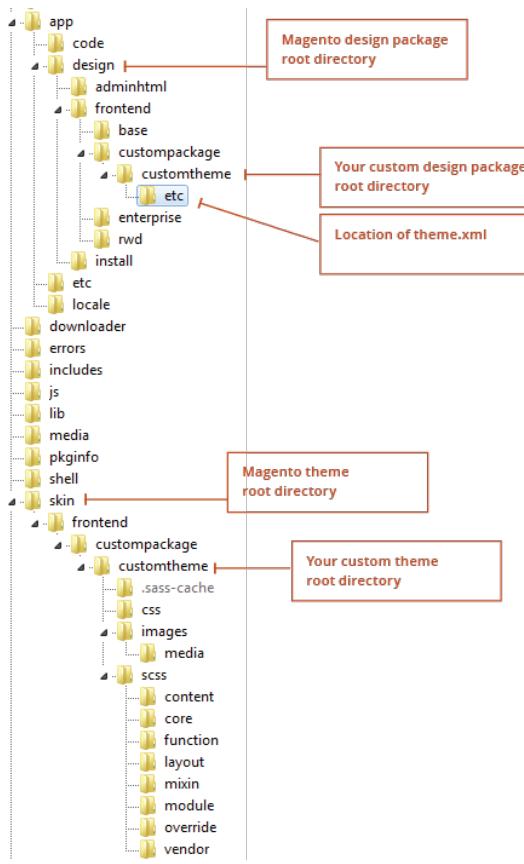
- `styles.css` loads from the correct directory
- Some HTTP 403 (Forbidden) errors loading images
- Some HTTP 404 (Not Found) errors loading images from `skin/frontend/base/default/images/media` (which doesn't exist)

*Solution:* Review the information discussed in [Step 2: Copying Files For Your Theme](#).

Make sure `theme.xml` for CE or EE is configured *exactly* as shown [here](#).

## Directory and File Reference

The following figure shows the directory structure in a properly configured system, as well as the location of `theme.xml`, the custom theme configuration file.



## Responsive Theme Reference

This section discusses information you need to customize the responsive theme you started in [Copying Files to Create Your Custom Theme](#).

See one of the following sections for more information:

- [Basic Recommendations for Development](#)
- [Understanding the Sass Structure](#)
- [Adding Your Own Sass Partials](#)
- [Using Media Query Breakpoints](#)
- [Working With Images](#)
- [Working With JavaScript](#)

## Basic Recommendations for Development

We recommend you do the following things when you start developing your responsive theme:

- Edit [your Magento install dir]/skin/frontend/custompackage/customtheme/scss/config.rb to change environment = :production to environment = :development.  
For more information about this setting, see [Compass Configuration Reference](#) .
- Disable the Magento cache as discussed in [Disabling Magento's Cache](#) .

## Understanding the Sass Structure

In previous versions of Magento, all CSS styles were grouped into a massive `styles.css` file that was cumbersome and difficult to navigate. The CSS in the responsive theme is organized into individual files referred to as *Sass partials*.

When Compass compiles the Sass partials, the output is a set of CSS files that are loaded by the browser: `styles-ie8.css` is for Internet Explorer 8 and earlier, and `styles.css` is for all other browsers.

Following is the directory structure of the `skin/frontend/custompackage/customtheme` directory.



**Note:** This structure does not include the sample data assets that you will remove later in this guide.

```

css
 +-- scaffold-forms.css
 +-- styles-ie8.css
 +-- styles.css

```

```

└── images
| ├── ajax-loader.gif
| ├── bg_x.png
| ├── demo-logo.png
| ├── demo_logo.png
| ├── icon_sprite.png
| ├── icon_sprite@2x.png
| ├── logo.png
| ├── logo_small.png
| ├── opc-ajax-loader.gif
| ├── social_icons.png
| └── social_icons@2x.png
└── scss
 ├── _core.scss
 ├── _framework.scss
 ├── _var.scss
 ├── config.rb
 ├── content
 | ├── _category.scss
 | ├── _home.scss
 ├── core
 | ├── _common.scss
 | ├── _form.scss
 | ├── _reset.scss
 | └── _table.scss
 ├── function
 | ├── _black.scss
 | └── _white.scss
 ├── layout
 | ├── _footer.scss
 | ├── _global.scss
 | ├── _header-account.scss
 | ├── _header-cart.scss
 | ├── _header-nav.scss
 | ├── _header-search.scss
 | └── _header.scss
 ├── mixin
 | ├── _breakpoint.scss
 | ├── _clearfix.scss
 | ├── _if-resolution.scss
 | ├── _image-replacement.scss
 | ├── _loading-overlay.scss
 | ├── _menu.scss
 | ├── _not-selectable.scss
 | ├── _toggle-content.scss
 | ├── _triangle.scss
 | └── _typography.scss
 ├── module
 | ├── _account-reviews.scss
 | ├── _catalog-compare.scss
 | ├── _catalog-msrp.scss
 | ├── _catalog-product.scss
 | ├── _checkout-cart-minicart.scss
 | ├── _checkout-cart.scss
 | ├── _checkout-onepage.scss
 | ├── _checkout-success.scss
 | ├── _cms.scss
 | ├── _configurableswatches.scss
 | ├── _contacts.scss
 | ├── _cookies.scss
 | ├── _customer.scss
 | ├── _paypal.scss
 | ├── _pricing_conditions.scss
 | ├── _product-list.scss
 | ├── _review.scss
 | ├── _search.scss
 | ├── _slideshow.scss
 | └── _wishlist.scss
 ├── override
 | └── _plugin.scss
 ├── scaffold-forms.scss
 ├── styles-ie8.scss
 ├── styles.scss
 └── vendor
 └── _normalize.scss

```

This is an explanation of the files in the preceding list:

- `config.rb`: Configuration values that tell Compass to look in the `scss` directory for Sass files and to export the CSS to the `css` directory. It also includes information about how Compass should format the compiled CSS.

- `scss/styles.scss`: Wrapper that imports the `framework.scss` and `core.scss` files.
- `scss/styles-ie8.scss`: Identical to `styles.css` file, except it does not include any media queries because Internet Explorer 8 and earlier don't support media queries. For more information, see [Using the `bp\(\)` mixin for older Internet Explorer Browser Versions](#).
- `scss/_framework.scss`: Imports all of the libraries, variables, functions, and mixins used by the other stylesheets. The files imported by `framework.scss` do not output any CSS on their own. Because of that, this file can be imported by other Sass files that are intended to compile directly to CSS files.  
For example, in Magento EE, the `skin/frontend/rwd/enterprise/scss/enterprise.scss` file imports `skin/frontend/rwd/default/scss/_framework.scss` so the Magento EE partials can use the common variables, mixins, and so on imported by `framework.scss`.
  - `scss/_var.scss`: Contains variables like breakpoints, spacing, colors, and font styles that are used by the other Sass partials. You should expect to modify this file heavily as you customize your responsive theme.
  - `scss/mixins/`: This directory contains all of the mixins used by the responsive theme. You should familiarize yourself with these mixins because they are useful for writing your custom styles.
  - `scss/functions/`: This directory contains functions that the responsive theme uses. Functions are mixins that return simple values.
- `scss/_core.scss`: Imports all of the Sass partials that output CSS content:
  - `scss/core/`: Partial that contain all of the global styles. Expect to heavily customize the partials in this directory.
  - `scss/layout/`: Partial that define the structural layout of the responsive theme (header, content, and footer). The `scss/core/_common.scss` partial also contains some layout styles—specifically the one, two, or three column layouts.
  - `scss/module/`: Partial that contain styles grouped by module. While the correlation between these module files and Magento core modules is close, it is not a 1:1 relationship. For example, the `_product-list.scss` file contains styles for both the `Mage_Catalog` and `Mage_CatalogSearch` modules.
  - `scss/vendor/`: This directory is intended for third-party CSS libraries like the `_normalize.scss` partial that normalizes how browsers render elements.
- `scss/scaffold-forms.scss`: For more information, see [Scaffold Forms](#).

## Adding Your Own Sass Partial

To customize styles for an existing Magento theme, you should edit existing Sass partials. However, if you are building a highly custom theme, you will likely want to create new partial files to contain styles for new functionality that you are introducing. Here is how to do that:

1. Add your partials to a new file named `skin/frontend/custompackage/customtheme/scss/module/_new-module.scss`.
2. Create a new `skin/frontend/custompackage/customtheme/scss/_custom_core.scss` file with the following line to import your new partial:

```
@import "module/new-module";
```

**Note:** Although you could edit `core.scss` to import your new partials, you should create `_custom_core.scss` to clearly distinguish which partial files are custom versus native.

3. Add this line to the bottom of `styles.scss` and `styles-ie8.scss` files:

```
@import "custom_core";
```

The contents of your new Sass partial will now be included in the output CSS files.

## Using Media Query Breakpoints

The responsive theme has a set of [breakpoints](#) it uses consistently (except for a few one-off breakpoints). These breakpoints are defined in both `_var.scss` and `app.js` so the CSS and JavaScript functionality stay in sync:

```
$bp-xsmall: 479px;
$bp-small: 599px;
$bp-medium: 770px;
$bp-large: 979px;
$bp-xlarge: 1199px;
```

A custom `bp()` mixin includes media query breakpoints.

For example, this Sass:

```
@include bp(max-width, $bp-medium) {
 /* These styles will only display on viewports smaller than or equal to 770px */
}
@include bp(min-width, $bp-medium+1) {
 /* These styles will only display on viewports larger than or equal to 771px (770px + 1px) */
}
```

generates this CSS:

```
@media only screen and (max-width: 770px) {
 /* These styles will only display on viewports smaller than or equal to 770px */
}
@media only screen and (min-width: 771px) {
 /* These styles will only display on viewports larger than or equal to 771px (770px + 1px) */
}
```

The breakpoint variables are intended to be used with `max-width` by default. When you use the breakpoint with `min-width`, you should add a pixel (`$bp-medium+1`) so the `min-width` and `max-width` styles don't both get applied to the same viewport size.

## Using the `bp()` mixin for older Internet Explorer Browser Versions

It is important to use the `bp()` mixin for your media query breakpoints because Internet Explorer 8 and earlier do not support media queries. While you can use a polyfill like [respond.js](#) to add media query support to IE versions 6–8, doing so will impact performance. Additionally, using a polyfill doesn't add much value because nearly all IE 6–8 users view web pages on a large screen device (1024px or larger), so those users won't benefit from media queries anyway.

To address the fact that IE 6–8 doesn't support media queries, the `styles-ie8.scss` stylesheet defines a couple of variables:

```
$mq-support: false;
$mq-fixed-value: 1024px;
```

With those variables defined, the `bp()` mixin outputs only the styles that would be visible on a 1024px+ viewport *and* it does not wrap those styles in a media query so that the IE browsers always render those styles. Refer to [\\_breakpoint.scss](#) to see exactly how this is done.

While it is acceptable to use the breakpoint variables defined by the responsive theme, when building your own responsive theme, your breakpoints should be specifically tailored to the content of your site. For example, let's say your main navigation only has four links and it fits nicely in a horizontal layout on viewports larger than 400px.

The responsive theme changes the horizontal layout to a collapsed drop-down on viewports smaller than 771px. Rather than defaulting to how the theme collapses the navigation, you should change the navigation to break at about 400px so users on screens between 400px and 771px get to experience the benefit of viewing the horizontal navigation.

## Working With Images

There are very few images in the responsive theme compared to the other Magento themes because CSS3 is used whenever possible to accomplish a visual effect.

### Working With the Header Logo

The responsive theme enables you to specify two different logos: one optimized for small screens (narrower than 770px) and the other for large screens. Place your two logo files in `skin/frontend/custompackage/customtheme/images` and specify the path to both logos in the Magento Admin Panel:

1. Log in to the Magento Admin Panel as an administrator.
2. Click **System > Configuration > GENERAL > Design**.
3. In the right pane, click **Header** to expand it.
4. In the **Logo Image Src** field, enter the path of your primary logo. For example: `images/logo.png`.
5. In the **Small Logo Image Src** field, enter the path of your logo for small viewports. For example: `images/logo_small.png`.
6. In the top right corner of the page, click **Save Config**.
7. If prompted, flush the cache:
  - a. Click **System > Configuration > Cache Management**.
  - b. At the top of the page, click **Flush Magento Cache**.

## Working With Sprites

The responsive theme uses two sprites: one for general icons and the other for social icons.

Both sprite images output at 1x and 2x resolutions (for high-resolution screens). Because sprite files are very difficult to edit without the source files, you should use these Photoshop files to change these sprites: [RWD\\_icon\\_sprite.psd](#) and [RWD\\_social\\_icons.psd](#)

Compass provides support for generating sprites from individual image files, but it was not used to generate sprites for the responsive theme. However, you might want to take advantage of that feature for your custom theme. Using this method requires knowledge of Compass.

## Working With JavaScript

All of the custom JavaScript in the responsive theme is contained in skin/frontend/rwd/default/js/app.js. To customize the responsive theme's JavaScript, copy app.js to skin/frontend/custompackage/customtheme/js and edit it there.

All custom JavaScript is based on jQuery (v1.10.2). In addition to jQuery, the following JavaScript libraries are included in the responsive theme:

- [enquire.js](#) : Used to run JavaScript conditionally, based on viewport size.
- [imagesloaded.js](#) : Used to determine whether images are fully loaded on the product detail page before activating zoom.
- [jquery.cycle2.min.js](#) : Powers the rotating banners on the homepage (requires Magento's sample data). This library can be removed if you aren't using the sample data. If you want to implement slide shows, [Owl Carousel](#) is recommended over Cycle2, as it has true swipe support and has more options.
- [jquery.cycle2.swipe.min.js](#): Adds pseudo-swipe support to Cycle2.
- [jquery-1.10.2.min.js](#) : jQuery library
- [matchMedia.js](#) and [matchMedia.addListener.js](#) : Used by enquire.js to provide support for Internet Explorer 9.
- [modernizr.custom.min.js](#) : Adds classes to the <html> tag to indicate whether browser supports certain CSS3 features and whether device is touch-enabled. The .touch and .no-touch classes are used in the Sass partials. They provide a Modernizr.mq function to test a media query in a cross-browser manner.
- [selectivizr.js](#) : Adds support for CSS3 selectors in IE 6–8
- [elevatezoom/jquery.elevateZoom-3.0.8.min.js](#) : Powers image zoom on product detail pages.

## Removing Sample Data Assets

The responsive theme includes several assets that are intended to power Magento's sample data. We recommend you remove them before you build your custom theme.

1. Remove the following files and directories:

```
skin/frontend/custompackage/customtheme/images/media
skin/frontend/custompackage/customtheme/scss/madisonisland.scss
skin/frontend/custompackage/customtheme/scss/madisonisland-ie8.scss
skin/frontend/custompackage/customtheme/scss/content/_category.scss
skin/frontend/custompackage/customtheme/scss/content/_home.scss
skin/frontend/custompackage/customtheme/css/madisonisland.css
skin/frontend/custompackage/customtheme/css/madisonisland-ie8.css
```

2. In your custom layout file (for example, app/design/frontend/custompackage/customtheme/layout/local.xml), add this XML:

```
<layout version="0.1.0">
 <default>
 <reference name="head">
 <action method="removeItem"><type>skin_js</type><name>js/slideshow.js</name></action>
 <action method="removeItem"><type>skin_js</type><name>js/lib/jquery.cycle2.min.js</name></acti
on>
 <action method="removeItem"><type>skin_js</type><name>js/lib/jquery.cycle2.swipe.min.js</name>
 </action>
 <action method="removeItem"><type>skin_css</type><name>css/madisonisland.css</name></actio
n>
 <action method="removeItem"><type>skin_css</type><name>css/madisonisland-ie8.css</name></actio
n>
 <action method="removeItem"><type>link_rel</type><name>//fonts.googleapis.com/css?family=Ralew
ay:300,400,500,700,600</name></action>
 </reference>
 </default>
</layout>
```

The preceding XML:

- Removes the Madison Island CSS files
- Removes the Raleway font that is used as a part of the default theme
- Removes the jQuery Cycle plugin and the slideshow.js file that activates Cycle.

3. Update this line in skin/frontend/custompackage/customtheme/scss/\_var.scss to reference whatever font you want to use for your custom site font:

```
$f-stack-special: 'Raleway', 'Helvetica Neue', Verdana, Arial, sans-serif;
```

If you want to use a custom web font for your theme (such as a [Google Font](#)), add XML like this to the <reference name="head"> section of layout.xml (see step 3). Customize the <href> value to the value that your custom web font provider gives you. It is recommended to include the same font weights that are referenced below (300,400,500,700,600) because the responsive theme uses each of those font weights.

```
<action method="addLinkRel"><rel>stylesheet</rel><href>//fonts.googleapis.com/css?family=Raleway:300,400,500,700,600</href></action>
```

## Advanced Topics

After you get familiar with the new structure, we encourage you to read these topics to fully take advantage of responsive web design.

### Getting Your Theme Ready for Production

To prepare your responsive theme for production, you should update `config.rb` in your theme directory to use these values:

```
output_style = :compressed
environment = :production
```

After you do this, you should recompile all of your Sass files—you can do this on the command line by running these commands inside your theme directory: `compass clean` and then `compass compile`. Finally, deploy the compiled CSS to your production system.

### Sass Fallback Structure

We recommend copying all of the Sass files from `skin/frontend/rwd/default/scss` to `skin/frontend/custompackage/customtheme/scss` to make it easy to get started with custom theme development. The downside of this approach is that it results in duplicating many files that you never edit directly.

In the end, your custom theme must include all Sass partials you want to override. There are at least two ways to do this:

- If you followed the setup instructions in [Step 1: Creating a Directory Structure for Your Responsive Theme](#), you must delete some files from your custom theme.

Delete the following directories to remove all the Sass partials that you copied from `rwd/default`:

```
skin/frontend/custompackage/customtheme/scss/core/
skin/frontend/custompackage/customtheme/scss/function/
skin/frontend/custompackage/customtheme/scss/layout/
skin/frontend/custompackage/customtheme/scss/mixin/
skin/frontend/custompackage/customtheme/scss/module/
skin/frontend/custompackage/customtheme/scss/override/
skin/frontend/custompackage/customtheme/scss/vendor/
```

The following files should now be located in `skin/frontend/custompackage/customtheme/scss`:

```
_var.scss
_core.scss
styles-ie8.scss
styles.scss
scaffold-forms.scss
_custom_core.scss
_framework.scss
```

- If you're familiar with Magento theme development and you did *not* already copy Sass partials, copy *only* the Sass partials you want to override from `rwd/default` to `skin/frontend/custompackage/customtheme/scss`.

Now you can set up a fallback structure:

1. Add the following line to `skin/frontend/rwd/default/config.rb` (omit the first line if you're using CE):

```
add_import_path "../../rwd/enterprise/scss"
add_import_path "../../rwd/default/scss"
```

The preceding causes Compass to look for Sass files in `skin/frontend/rwd/enterprise/scss` first and then in `skin/frontend/rwd/default/scss` if they can't be found in `skin/frontend/custompackage/customtheme/scss`.

2. Following is what to do with copies of Sass files from `rwd/default`:

- `styles.scss`, `styles-ie8.scss`: You must copy these files into your custom theme so that Compass knows which files to compile into CSS files.
- `_core.scss`, `framework.scss`: You must copy these files into your custom theme so Compass looks in the `custompackage/customtheme` directory for any of the files imported by these two files.
- `_var.scss`: You'll typically edit the values in this file for your custom theme.  
If you don't like the idea of editing this file and want to override its values with a `_var_custom.scss` file, make sure it gets imported immediately after `_var.scss`.

3. As a reminder, to override Sass partials in `rwd/default`, you must copy them into your custom theme so they'll get included by Compass.

**Note:** If you're using `compass watch` to recompile your Sass, you must stop and restart Compass any time you copy a new Sass file into your custom theme. Otherwise, the Compass compiler won't know about the existence of the new file.

To avoid having to stop and restart `compass watch`, start your project by copying all Sass files from `rwd/default` into your custom

theme, and then delete all unchanged Sass files at the end of your project.

## Scaffold Forms

The Magento responsive theme ships with `skin/frontend/rwd/default/scss/scaffold-forms.scss` that has styles that cause the labels of long forms (for example, the Billing Information step of checkout) to display to the left of their associated input on larger viewports (as opposed to on top). This layout is intended to make long forms seem less intimidating to users.

However, these styles are not currently implemented because the Magento responsive theme must work well with internationalization (where the length of labels can vary greatly, thereby impacting the scaffold form layout).

It's up to you whether or not to use these styles. To do so, move `skin/frontend/rwd/default/scss/scaffold-forms.scss` to `core/_scaffold-forms.scss` and edit `scss/_core.scss` to include it.

## High-Definition Images

The responsive theme has no high-resolution image solution for product listing pages. Refer to [slides starting at 114](#) of Brendan Falkowski's Imagine 2013 talk for implementation suggestions. For a 2x solution, PictureFill (or similar polyfill) could be used, although it would need to be integrated with ElevateZoom.



**Note:** Category banners and product detail pages use a 1.5x image solution.

Magento does not resize the main product photo on product detail pages, so you must implement high-resolution support in your responsive theme. You must be careful with this if you have very large product photos because it could drastically impact performance.

To constrain the size of your main product photo, open `app/design/frontend/rwd/default/template/catalog/product/view/media.phtml` and modify this line:

```
<?php echo $this->helper('catalog/image')->init($_product, 'image') ?>
```

to something like:

```
src="<?php echo $this->helper('catalog/image')->init($_product, 'image')->constrainOnly(true)->resize(1800); ?>"
```

## Getting Help With Your Theme

If you encounter issues building a custom responsive theme, you can post your question on the community [magento.stackexchange.com](#) site and tag it with `rwd-theme`.

To browse the list of questions, see <http://magento.stackexchange.com/questions/tagged/rwd-theme>.

22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Creating an iOS App for Magento MobileConnect

The iOS app for Magento MobileConnect was discontinued in 2015.

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento for Developers: Part 2—The Magento Config

by Alan Storm, updated for Magento 1.12

[Edit this page on GitHub](#) 

### Other articles in this series:

- [Part 1—Introduction to Magento](#)
- [Part 2—The Magento Config](#)
- [Part 3—Magento Controller Dispatch](#)
- [Part 4—Magento Layouts, Blocks and Templates](#)
- [Part 5—Magento Models and ORM Basics](#)
- [Part 6—Magento Setup Resources](#)
- [Part 7—Advanced ORM: Entity Attribute Value](#)
- [Part 8—Varien Data Collections](#)

The config is the beating heart of the Magento System. It describes, in whole, almost any module, model, class, template, etc. than you'll need to access. It's a level of abstraction that most PHP developers aren't used to working with, and while it adds development time in the form of confusion and head scratching, it also allows you an unprecedented amount of flexibility as far as overriding default system behaviors go.

To start with, we're going to create a Magento module that will let us view the system config in our web browser. Follow along by copying and pasting the code below, it's worth going through on your own as a way to start getting comfortable with things you'll be doing while working with Magento, as well as learning key terminology.

### In this article...

- [Setting up a Module Directory Structure](#)
- [Creating a Module Config](#)
- [What Am I Looking at?](#)
- [Why Do I Care?](#)

## Setting up a Module Directory Structure

We're going to be creating a Magento module. A module is a group of php and xml files meant to extend the system with new functionality, or override core system behavior. This may mean adding additional data models to track sales information, changing the behavior of existing classes, or adding entirely new features.

It's worth noting that most of the base Magento system is built using the same module system you'll

be using. If you look in

```
app/code/core/Mage
```

each folder is a separate module built by the Magento team. Together, these modules form the community shopping cart system you're using. Your modules should be placed in the following folder

```
app/code/local/Packagename
```

"Packagename" should be a unique string to Namespace/Package your code. It's an unofficial convention that this should be the name of your company. The idea is to pick a string that no one else in the world could possibly be using.

```
app/code/local/Microsoft
```

We'll use "Magentotutorial".

So, to add a module to your Magento system, create the following directory structure

```
app/code/local/Magentotutorial/Configviewer/Block
app/code/local/Magentotutorial/Configviewer/controllers
app/code/local/Magentotutorial/Configviewer/etc
app/code/local/Magentotutorial/Configviewer/Helper
app/code/local/Magentotutorial/Configviewer/Model
app/code/local/Magentotutorial/Configviewer/sql
```

You won't need all these folder for every module, but setting them all up now is a smart idea.

Next, there's two files you'll need to create. The first, config.xml, goes in the etc folder you just created.

```
app/code/local/Magentotutorial/Configviewer/etc/config.xml
```

The second file should be created at the following location

```
app/etc/modules/Magentotutorial_Configviewer.xml
```

The naming convention for this files is Packagename\_Modulename.xml.

The config.xml file should contain the following XML. Don't worry too much about what all this does for now, we'll get there eventually

```
<config>
 <modules>
 <Magentotutorial_Configviewer>
 <version>0.1.0</version>
 </Magentotutorial_Configviewer>
 </modules>
</config>
```

Finally, Magentotutorial\_Configviewer.xml should contain the following xml.

```
<config>
 <modules>
 <Magentotutorial_Configviewer>
 <active>true</active>
 <codePool>local</codePool>
 </Magentotutorial_Configviewer>
 </modules>
</config>
```

That's it. You now have a bare bones module that won't do anything, but that Magento will be aware of. To make sure you've done everything right, do the following:

1. Clear your Magento cache
2. In the Magento Admin, go to **System->Configuration->Advanced**
3. In the "Disable modules output" panel verify that Magentotutorial\_Configviewer shows up

Congratulations, you've built your first Magento module!

## Creating a Module Config

Of course, this module doesn't do anything yet. When we're done, our module will

1. Check for the existence of a "showConfig" query string variable
2. If showConfig is present, display our Magento config and halt normal execution
3. Check for the existence of an additional query string variable, showConfigFormat that will let us specify text or xml output.

First, we're going to add the following <global> section to our config.xml file.

```
<config>
 <modules>...</modules>
 <global>
 <events>
 <controller_front_init_routers>
 <observers>
 <Magentotutorial_configviewer_model_observer>
 <type>singleton</type>
 <class>Magentotutorial_Configviewer_Model_Observer</c
lass>
 <method>checkForConfigRequest</method>
 </Magentotutorial_configviewer_model_observer>
 </observers>
 </controller_front_init_routers>
 </events>
 </global>
</config>
```

Then, create a file at

```
Magentotutorial/Configviewer/Model/Observer.php
```

and place the following code inside

```
<?php
 class Magentotutorial_Configviewer_Model_Observer {
 const FLAG_SHOW_CONFIG = 'showConfig';
 const FLAG_SHOW_CONFIG_FORMAT = 'showConfigFormat';

 private $request;

 public function checkForConfigRequest($observer) {
 $this->request = $observer->getEvent()->getData('front')->getRequ
est();
 if($this->request->{self::FLAG_SHOW_CONFIG} === 'true'){
 $this->setHeader();
 $this->outputConfig();
 }
 }

 private function setHeader() {
 $format = isset($this->request->{self::FLAG_SHOW_CONFIG_FORMAT})?
 $this->request->{self::FLAG_SHOW_CONFIG_FORMAT} : 'xml';
 switch($format){
 case 'text':
 header("Content-Type: text/plain");
 break;
 default:
 header("Content-Type: text/xml");
 }
 }

 private function outputConfig() {
 die(Mage::app()->getConfig()->getNode()->asXML());
 }
 }
}
```

That's it. Clear your Magento cache again and then load any Magento URL with a showConfig=true query string

```
http://magento.example.com/?showConfig=true
```

What am I looking at?

You should be looking at a giant XML file. This describes the state of your Magento system. It lists all modules, models, classes, event listeners or almost anything else you could think of.

For example, consider the config.xml file you created above. If you search the XML file in your browser for the text `Configviewer_Model_Observer` you'll find your class listed. Every module's config.xml file is parsed by Magento and included in the global config.

## Why Do I Care?

Right now this may seem esoteric, but this config is key to understanding Magento. Every module you'll be creating will add to this config, and anytime you need to access a piece of core system functionality, Magento will be referring back to the config to look something up.

A quick example: As an MVC developer, you've likely worked with some kind of helper class, instantiated something like

```
$helper_sales = new HelperSales();
```

One of the things Magento has done is abstract away PHP's class declaration. In Magento, the above code looks something like

```
$helper_sales = Mage::helper('sales');
```

In plain english, the static helper method will:

1. Look in the `<helpers />` section of the Config.
2. Within `<helpers />`, look for a `<sales />` section
3. Within the `<sales />` section look for a `<class />` section
4. Append the part after the slash to the value found in #3 (defaulting to `data` in this case)
5. Instantiate the class found in #4 (`Mage_Sales_Helper_Data`)

While this seems like a lot of work (and it is), the key advantage is by always looking to the config file for class names, we can override core Magento functionality **without** changing or adding to the core code. This level of meta programming, not usually found in PHP, allows you to cleanly extend only the parts of the system you need to.

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento for Developers: Part 3—Magento Controller Dispatch

by [Alan Storm](#), updated for Magento 1.12

[Edit this page on GitHub](#) 

### Other articles in this series:

- [Part 1—Introduction to Magento](#)
- [Part 2—The Magento Config](#)
- [Part 3—Magento Controller Dispatch](#)
- [Part 4—Magento Layouts, Blocks and Templates](#)
- [Part 5—Magento Models and ORM Basics](#)
- [Part 6—Magento Setup Resources](#)
- [Part 7—Advanced ORM: Entity Attribute Value](#)
- [Part 8—Varien Data Collections](#)

The Model-View-Controller (MVC) architecture traces its origins back to the Smalltalk Programming language and Xerox Parc. Since then, there have been many systems that describe their architecture as MVC. Each system is slightly different, but all have the goal of separating data access, business logic, and user-interface code from one another.

The architecture of most PHP MVC frameworks will look something [like this](#).

1. A URL is intercepted by a single PHP file (usually called a Front Controller).
2. This PHP file will examine the URL, and derive a Controller name and an Action name (a process that's often called routing).
3. The derived Controller is instantiated.
4. The method name matching the derived Action name is called on the Controller.
5. This Action method will instantiate and call methods on models, depending on the request variables.
6. The Action method will also prepare a data structure of information. This data structure is passed on to the view.
7. The view then renders HTML, using the information in the data structure it has received from the Controller.

While this pattern was a great leap forward from the "each php file is a page" pattern established early on, for some software engineers, it's still an ugly hack. Common complaints are:

- The Front Controller PHP file still operates in the global namespace.

- Convention over configuration leads to less modularity.
- URLs routing is often inflexible.
- Controllers are often bound to specific views.
- Even when a system offers a way to override these defaults, the convention leads to applications where it's difficult/impossible to drop in a new model, view, or Controller implementation without massive re-factoring.

As you've probably guessed, the Magento team shares this world view and has created a more abstract MVC pattern that looks something [like this](#):

1. A URL is intercepted by a single PHP file.
2. This PHP file instantiates a Magento application.
3. The Magento application instantiates a Front Controller object.
4. Front Controller instantiates any number of Router objects (specified in global config).
5. Routers check the request URL for a "match".
6. If a match is found, an Action Controller and Action are derived.
7. The Action Controller is instantiated and the method name matching the Action Name is called.
8. This Action method will instantiate and call methods on models, depending on the request.
9. This Action Controller will then instantiate a Layout Object.
10. This Layout Object will, based some request variables and system properties (also known as "handles"), create a list of Block objects that are valid for this request.
11. Layout will also call an output method on certain Block objects, which start a nested rendering (Blocks will include other Blocks).
12. Each Block has a corresponding Template file. Blocks contain PHP logic, templates contain HTML and PHP output code.
13. Blocks refer directly back to the models for their data. In other words, **the Action Controller does not pass them a data structure**.

We'll eventually touch on each part of this request, but for now we're concerned with the **Front Controller -> Routers -> Action Controller** section.

## Hello World

Enough theory, it's time for Hello World. We're going to

1. Create a Hello World module in the Magento system
2. Configure this module with routes
3. Create Action Controller(s) for our routes

### Create Hello World Module

First, we'll create a directory structure for this module. Our directory structure should look as follows:

```
app/code/local/Magentotutorial/Helloworld/Block
app/code/local/Magentotutorial/Helloworld/controllers
app/code/local/Magentotutorial/Helloworld/etc
app/code/local/Magentotutorial/Helloworld/Helper
app/code/local/Magentotutorial/Helloworld/Model
app/code/local/Magentotutorial/Helloworld/sql
```

Then create a configuration file for the module (at path

`app/code/local/Magentotutorial/Helloworld/etc/config.xml`):

```
<config>
<modules>
 <Magentotutorial_Helloworld>
 <version>0.1.0</version>
 </Magentotutorial_Helloworld>
</modules>
</config>
```

Then create a file to activate the module (at path

```
app/etc/modules/Magentotutorial_Helloworld.xml):
```

```
<config>
 <modules>
 <Magentotutorial_Helloworld>
 <active>true</active>
 <codePool>local</codePool>
 </Magentotutorial_Helloworld>
 </modules>
</config>
```

Finally, we ensure the module is active:

1. Clear your Magento cache.
2. In the Magento Admin, go to **System->Configuration->Advanced**.
3. Expand "Disable Modules Output" (if it isn't already).
4. Ensure that Magentotutorial\_Helloworld shows up.

## Configuring Routes

Next, we're going to configure a route. A route will turn a URL into an Action Controller and a method. Unlike other convention based PHP MVC systems, with Magento you need to explicitly define a route in the global Magento config.

In your config.xml file(at path app/code/local/Magentotutorial/Helloworld/etc/config.xml), add the following section:

```
<config>
 ...
 <frontend>
 <routers>
 <helloworld>
 <use>standard</use>
 <args>
 <module>Magentotutorial_Helloworld</module>
 <frontName>helloworld</frontName>
 </args>
 </helloworld>
 </routers>
 </frontend>
 ...
</config>
```

We have a lot of new terminology here, let's break it down.

### What is a <frontend>?

The <frontend> tag refers to a Magento Area. For now, think of Areas as individual Magento applications. The "frontend" Area is the public facing Magento shopping cart application. The "admin" Area is the private administrative console application. The "install" Area is the application you use to run through installing Magento the first time.

### Why a <routers> tags if we're configuring individual routes?

There's a famous quote about computer science, often attributed to Phil Karlton:

"There are only two hard things in Computer Science: cache invalidation and naming things"

Magento, like all large systems, suffers from the naming problem in spades. You'll find there are many places in the global config, and the system in general, where the naming conventions seem unintuitive or even ambiguous. This is one of those places. Sometimes the <routers> tag will enclose configuration information about routers, other times it will enclose configuration information about the actual router objects that do the routing. This is going to seem counter intuitive at first, but as you start to work with Magento more and more, you'll start to understand its world view a little better. (Or, in the words of Han Solo, "Hey, trust me!").

### What is a <frontName>?

When a router parses a URL, it gets separated as follows

```
http://example.com/frontName/actionControllerName/actionMethod/
```

So, by defining a value of "helloworld" in the <frontName> tags, we're telling Magento that we want the system to respond to URLs in the form of

```
http://example.com/helloworld/*
```

Many developers new to Magento confuse this frontName with the Front Controller object. They are **not** the same thing. The frontName belongs solely to routing.

### What's the <helloworld> tag for?

This tag should be the lowercase version of your module name. Our module name is Helloworld, this tag is helloworld. Technically this tag defines our *route name*.

You'll also notice our frontName matches our module name. It's a loose convention to have frontNames match the module names, but it's not a requirement. In your own modules, it's probably better to use a route name that's a combination of your module name and package name to avoid possible namespace collisions.

### What's <module>Magentotutorial\_Helloworld</module> for?

This module tag should be the full name of your module, including its package/namespace name. This will be used by the system to locate your Controller files.

## Create Action Controller(s) for our Routes

One last step to go, and we'll have our Action Controller. Create a file at

```
app/code/local/Magentotutorial/Helloworld/controllers/IndexController.php
```

That contains the following

```
<?php
class Magentotutorial_Helloworld_IndexController extends Mage_Core_Controller
_Front_Action {
 public function indexAction() {
 echo 'Hello World';
 }
}
```

Clear your config cache, and load the following URL

```
http://example.com/helloworld/index/index
```

You should also be able to load

```
http://example.com/helloworld/index/
http://example.com/helloworld/
```

You should see a blank page with the text "Hello World". Congratulations, you've setup your first Magento Controller!

### Where do Action Controllers go?

Action Controllers should be placed in a module's controllers (lowercase c) folder. This is where the system will look for them.

### How should Action Controllers be named?

Remember the <module> tag back in config.xml?

```
<module>Magentotutorial_Helloworld</module>
```

An Action Controller's name will

1. Start with this <module> string specified in config.xml (Magentotutorial\_Helloworld)
2. Be followed by an underscore (Magentotutorial\_Helloworld\_)
3. Which will be followed by the Action Controller's name (Magentotutorial\_Helloworld\_Index)
4. And finally, the word "Controller" (Magentotutorial\_Helloworld\_IndexController)

All Action Controllers need Mage\_Core\_Controller\_Front\_Action as an ancestor.

### What's that index/index nonsense?

As we previously mentioned, Magento URLs are routed (by default) as follows

```
http://example.com/frontName/actionControllerName/actionMethod/
```

So in the URL

```
http://example.com/helloworld/index/index
```

the URI portion "helloworld" is the frontName, which is followed by index (The Action Controller name), which is followed by another index, which is the name of the Action Method that will be called. (an Action of index will call the method `public function indexAction() { ... }`).

If a URL is incomplete, Magento uses "index" as the default, which is why the following URLs are equivalent.

```
http://example.com/helloworld/index
http://example.com/helloworld
```

If we had a URL that looked like this

```
http://example.com/checkout/cart/add
```

Magento would

1. Consult the global config to find the module to use for the frontName checkout (`Mage_Checkout`)
2. Look for the cart Action Controller (`Mage_Checkout_CartController`)
3. Call the `addAction` method on the cart Action Controller

## Other Action Controller Tricks

Let's try adding a non-default method to our Action Controller. Add the following code to `IndexController.php`

```
public function goodbyeAction() {
 echo 'Goodbye World!';
}
```

And then visit the URL to test it out:

```
http://example.com/helloworld/index/goodbye
```

Because we're extending the `Mage_Core_Controller_Front_Action` class, we get some methods for free. For example, additional URL elements are automatically parsed into key/value pairs for us. Add the following method to your Action Controller.

```
public function paramsAction() {
 echo '<dl>';
 foreach($this->getRequest()->getParams() as $key=>$value) {
 echo '<dt>Param: '.$key.'.</dt>';
 echo '<dt>Value: '.$value.'.</dt>';
 }
 echo '</dl>';
}
```

and visit the following URL

```
http://example.com/helloworld/index/params?foo=bar&baz=eof
```

You should see each parameter and value printed out.

Finally, what would we do if we wanted a URL that responded at

```
http://example.com/helloworld/messages/goodbye
```

Here our Action Controller's name is messages, so we'd create a file at

```
app/code/local/Magentotutorial/Helloworld/controllers/MessagesController.php
```

with an Action Controller named `Magentotutorial_Helloworld_MessagesController` and an Action Method that looked something like

```
public function goodbyeAction()
{
 echo 'Another Goodbye';
}
```

And that, in a nutshell, is how Magento implements the Controller portion of MVC. While it's a little more complicated than other PHP MVC framework's, it's a highly flexible system that will allow you build almost any URL structure you want.

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento for Developers: Part 4—Magento Layouts, Blocks and Templates

by *Alan Storm*, updated for Magento 1.12

[Edit this page on GitHub](#) 

### Other articles in this series:

- [Part 1—Introduction to Magento](#)
- [Part 2—The Magento Config](#)
- [Part 3—Magento Controller Dispatch](#)
- [Part 4—Magento Layouts, Blocks and Templates](#)
- [Part 5—Magento Models and ORM Basics](#)
- [Part 6—Magento Setup Resources](#)
- [Part 7—Advanced ORM: Entity Attribute Value](#)
- [Part 8—Varien Data Collections](#)

Developers new to Magento are often confused by the Layout and View system. This article will take a look at Magento's Layout/Block approach, and show you how it fits into Magento MVC worldview.

Unlike many popular MVC systems, Magento's Action Controller does **not** pass a data object to the view or set properties on the view object (with a few exceptions). Instead, the View component directly references system models to get the information it needs for display.

One consequence of this design decision is that the View has been separated into Blocks and Templates. Blocks are PHP objects, Templates are "raw" PHP files (with a .phtml extension) that contain a mix of HTML and PHP (where PHP is used as a templating language). Each Block is tied to a single Template file. Inside a phtml file, PHP's `$this` keyword will contain a reference to the Template's Block object.

## A quick example

Take a look at the default product Template at the file at

`app/design/frontend/base/default/template/catalog/product/list.phtml`

You'll see the following PHP template code.

```
<?php $_productCollection=$this->getLoadedProductCollection() ?>
<?php if(!$_productCollection->count()): ?> <div class="note-msg">
```

```
<?php echo $this->__("There are no products matching the selection.") ?>
</div> <?php else: ?>
...

```

The `getLoadedProductCollection` method can be found in the Template's Block class, `Mage_Catalog_Block_Product_List` as shown:

File: `app/code/core/Mage/Catalog/Block/Product/List.php`

```
...
public function getLoadedProductCollection()
{
 return $this->_getProductCollection();
}
...

```

The block's `_getProductCollection` then instantiates models and reads their data, returning a result to the template.

## Nesting Blocks

The real power of Blocks/Templates come with the `getChildHtml` method. This allows you to include the contents of a secondary Block/Template inside of a primary Block/Template.

Blocks calling Blocks calling Blocks is how the entire HTML layout for your page is created. Take a look at the one column layout Template.

File: `app/design/frontend/base/default/template/page/1column.phtml`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->getLang() ?>" lang="<?php echo $this->getLang() ?>">
<head>
<?php echo $this->getChildHtml('head') ?>
</head>
<body class="page-popup <?php echo $this->getBodyClass()?$this->getBodyClass():'' ?>">
 <?php echo $this->getChildHtml('content') ?>
 <?php echo $this->getChildHtml('before_body_end') ?>
 <?php echo $this->getAbsoluteFooter() ?>
</body>
```

The template itself is only 28 lines long. However, each call to `$this->getChildHtml(...)` will include and render another Block. These Blocks will, in turn, use `getChildHtml` to render other Blocks. It's Blocks all the way down.

## The Layout

So, Blocks and Templates are all well and good, but you're probably wondering

1. How do I tell Magento which Blocks I want to use on a page?
2. How do I tell Magento which Block I should start rendering with?
3. How do I specify a particular Block in `getChildHtml(...)`? Those argument strings don't look like Block names to me.

This is where the Layout Object enters the picture. The Layout Object is an XML object that will define which Blocks are included on a page, and which Block(s) should kick off the rendering process.

[Last time](#) we were echoing content directly from our Action Methods. This time let's create a simple HTML template for our Hello World module.

First, create a file at

`app/design/frontend/base/default/layout/local.xml`

with the following contents

```
<layout version="0.1.0">
<default>
 <block type="page/html" name="root" output="toHtml" template="magento
```

```
tutorial/helloworld/simple_page.phtml" />
 </default>
</layout>
```

Then, create a file at

```
app/design/frontend/base/default/template/magentotutorial/helloworld/simple_p
age.phtml
```

with the following contents

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
 <title>Hello World</title>
 <style type="text/css">
 body {
 background-color:#f00;
 }
 </style>
</head>
<body>

</body>
</html>
```

Finally, each Action Controller is responsible for kicking off the layout process. We'll need to add two method calls to the Action Method.

```
public function indexAction() {
 //remove our previous echo
 //echo 'Hello Index!';
 $this->loadLayout();
 $this->renderLayout();
}
```

Clear your Magento cache and reload your Hello World controller page. You should now see a website with a bright red background and an HTML source that matches what's in simple\_page.phtml.

## What's Going On

So, that's a lot of voodoo and cryptic incantations. Let's take a look at what's going on.

First, you'll want to install the [Layoutviewer](#) module. This is a module similar to the [Configviewer](#) module you built in the Hello World article that will let us peek at some of Magento's internals.

Once you've installed the module (similar to how you setup the [Configviewer](#) module), go to the following URL

```
http://example.com/helloworld/index/index?showLayout=page
```

This is the layout XML for your page/request. It's made up of `<block />`, `<reference />` and `<remove />` tags. When you call the `loadLayout` method of your Action Controller, Magento will

1. Generate this Layout XML
2. Instantiate a Block class for each `<block />` tag, looking up the class using the tag's type attribute as a global config path and store it in the internal `_blocks` array of the layout object, using the tag's name attribute as the array key.
3. If the `<block />` tag contains an output attribute, its value is added to the internal `_output` array of the layout object.

Then, when you call the `renderLayout` method in your Action Controller, Magento will iterate over all the Blocks in the `_output` array, using the value of the output attribute as a callback method. This is always `toHtml`, and means the starting point for output will be that Block's Template.

The following sections will cover how Blocks are instantiated, how this layout file is generated, and finishes up with kicking off the output process.

## Block Instantiation

So, within a Layout XML file, a `<block />` has a "type" that's actually a Grouped Class Name URI

```
<block type="page/html" ...
<block type="page/template_links" ...
```

The URI references a location in the (say it with me) global config. The first portion of the URI (in the above examples `page`) will be used to query the global config to find the page class name. The second portion of the URI (in the two examples above, `html` and `template_links`) will be appended to the base class name to create the class name Magento should instantiate.

We'll go through `page/html` as an example. First, Magento looks for the global config node at file

```
app/code/core/Mage/Page/etc/config.xml
```

and finds

```
<page>
 <class>Mage_Page_Block</class>
</page>
```

This gives us our base class prefix `Mage_Page_Block`. Then, the second part of the URI (`html`) is appended to the class name to give us our final Block class name `Mage_Page_Block_Html`. This is the class that will be instantiated.

If we create a block with the same name as an already existing block, the new block instance will replace the original instance. This is what we've done in our `local.xml` file from above.

```
<layout version="0.1.0">
 <default>
 <block type="page/html" name="root" output="toHtml" template="magento
tutorial/helloworld/simple_page.phtml" />
 </default>
</layout>
```

The Block named `root` has been replaced with our Block, which points at a different phtml Template file.

## Using references

`<reference name="" />` will hook all contained XML declarations into an existing block with the specified name. Contained `<block />` nodes will be assigned as child blocks to the referenced parent block.

```
<layout version="0.1.0">
 <default>
 <block type="page/html" name="root" output="toHtml" template="page/2c
columns-left.phtml">
 <!-- ... sub blocks ... -->
 </block>
 </default>
</layout>
```

In a different layout file:

```
<layout version="0.1.0">
 <default>
 <reference name="root">
 <!-- ... another sub block ... -->
 <block type="page/someothertype" name="some.other.block.name" tem
plate="path/to/some/other/template" />
 </reference>
 </default>
</layout>
```

Even though the root block is declared in a separate layout XML file, the new block is added as a child block. Magento initially creates a `page/html` Block named `root`. Then, when it later encounters the reference with the same name (`root`), it will assign the new block `some.other.block.name` as a child of the root block.

## How Layout Files are Generated

So, we have a slightly better understanding of what's going on with the Layout XML, but where is this XML file coming from? To answer that question, we need to introduce two new concepts; Handles and the Package Layout.

## Handles

Each page request in Magento will generate several unique Handles. The Layoutview module can show you these Handles by using a URL something like

```
http://example.com/helloworld/index/index?showLayout=handles
```

You should see a list similar to the following (depending on your configuration)

1. default
2. STORE\_bare\_us
3. THEME\_frontend\_default\_default
4. helloworld\_index\_index
5. customer\_logged\_out

Each of these is a Handle. Handles are set in a variety of places within the Magento system. The two we want to pay attention to are `default` and `helloworld_index_index`. The `default` Handle is present in **every** request into the Magento system. The `helloworld_index_index` Handle is created by combining the route name (`helloworld`), Action Controller name (`index`), and Action Controller Action Method (`index`) into a single string. This means each possible method on an Action Controller has a Handle associated with it.

Remember that "index" is the Magento default for both Action Controllers and Action Methods, so the following request

```
http://example.com/helloworld/?showLayout=handles
```

Will also produce a Handle named `helloworld_index_index`

## Package Layout

You can think of the Package Layout similar to the global config. It's a large XML file that contains **every possible layout configuration** for a particular Magento install. Let's take a look at it using the Layoutview module

```
http://example.com/helloworld/index/index?showLayout=package
```

This may take a while to load. If your browser is choking on the XML rendering, try the text format

```
http://example.com/helloworld/index/index?showLayout=package&showLayoutFormat=text
```

You should see a very large XML file. This is the Package Layout. This XML file is created by combining the contents of all the XML layout files for the current theme (or package). For the default install, this is at

```
app/design/frontend/base/default/layout/
```

Behind the scenes there are `<frontend><layout><updates />` and `<adminhtml><layout><updates />` sections of the global config that contains nodes with all the file names to load for the respective area. Once the files listed in the config have been combined, Magento will merge in one last xml file, `local.xml`. This is the file where you're able to add your customizations to your Magento install.

## Combining Handles and The Package Layout

So, if you look at the Package Layout, you'll see some familiar tags such as `<block />` and `<reference />`, but they're all surrounded by tags that look like

```
<default />
<catalogsearch_advanced_index />
etc...
```

These are all Handle tags. The Layout for an individual request is generated by grabbing all the sections of the Package Layout that match any Handles for the request. So, in our example above, our layout is being generated by grabbing tags from the following sections

```
<default />
<STORE_bare_us />
<THEME_frontend_default_default />
<helloworld_index_index />
```

```
<customer_logged_out />
```

There's one additional tag you'll need to be aware of in the Package Layout. The `<update />` tag allows you to include another Handle's tags. For example

```
<customer_account_index>
<!-- ... -->
<update handle="customer_account"/>
<!-- ... -->
</customer_account_index>
```

Is saying that requests with a `customer_account_index` Handle should include `<blocks />`s from the `<customer_account />` Handle.

## Applying What We've Learned

OK, that's a lot of theory. Lets get back to what we did earlier. Knowing what we know now, adding

```
<layout version="0.1.0">
<default>
 <block type="page/html" name="root" output="toHtml" template="magento
tutorial/helloworld/simple_page.phtml" />
</default>
</layout>
```

to `local.xml` means we've overridden the `"root"` tag, with a different Block. By placing this in the `<default />` Handle we've ensured that this override will happen for **every page request** in the system. That's probably not what we want.

If you go to any other page in your Magento site, you'll notice they're either blank white, or have the same red background that your hello world page does. Let's change your `local.xml` file so it only applies to the hello world page. We'll do this by changing `default` to use the full action name handle (`helloworld_index_index`).

```
<layout version="0.1.0">
<helloworld_index_index>
 <block type="page/html" name="root" output="toHtml" template="magento
tutorial/helloworld/simple_page.phtml" />
</helloworld_index_index>
</layout>
```

Clear your Magento cache, and the rest of your pages should be restored.

Right now this only applies to our index Action Method. Let's add it to the goodbye Action Method as well. In your Action Controller, modify the goodbye action so it looks like

```
public function goodbyeAction() {
 $this->loadLayout();
 $this->renderLayout();
}
```

If you load up the following URL, you'll notice you're still getting the default Magento layout.

```
http://example.com/helloworld/index/goodbye
```

We need to add a Handle for the full action name (`helloworld_index_goodbye`) to our `local.xml` file. Rather than specify a new `<block />`, lets use the `update` tag to include the `helloworld_index_index` Handle.

```
<layout version="0.1.0">
<!-- ... -->
<helloworld_index_goodbye>
 <update handle="helloworld_index_index" />
</helloworld_index_goodbye>
</layout>
```

Loading the following pages (after clearing your Magento cache) should now produce identical results.

```
http://example.com/helloworld/index/index
http://example.com/helloworld/index/goodbye
```

## Starting Output and getChildHtml

In a standard configuration, output starts on the Block named root (because it has an output attribute). We've overridden root's Template with our own

```
template="magentotutorial/helloworld/simple_page.phtml"
```

Templates are referenced from the root folder of the current theme. In this case, that's

```
app/design/frontend/base/default
```

so we need to drill down to our custom page. Most Magento Templates are stored in

```
app/design/frontend/base/default/templates
```

Combining this gives us the full path

```
app/design/frontend/base/default/templates/magentotutorial/helloworld/simple_
page.phtml
```

## Adding Content Blocks

A simple red page is pretty boring. Let's add some content to this page. Change your `<helloworld_index_index />` Handle in local.xml so it looks like the following

```
<helloworld_index_index>
 <block type="page/html" name="root" output="toHtml" template="magentotutorial/helloworld/simple_page.phtml">
 <block type="customer/form_register" name="customer_form_register" template="customer/form/register.phtml"/>
 </block>
</helloworld_index_index>
```

We're adding a new Block nested within our root. This is a Block that's distributed with Magento, and will display a customer registration form. By nesting this Block within our root Block, we've made it available to be pulled into our `simple_page.html` Template. Next, we'll use the Block's `getChildHtml` method in our `simple_page.phtml` file. Edit `simple_page.html` so it looks like this

```
<body>
 <?php echo $this->getChildHtml('customer_form_register'); ?>
</body>
```

Clear your Magento cache and reload the page and you should see the customer registration form on your red background. Magento also has a Block named top.links. Let's try including that. Change your `simple_page.html` file so it reads

```
<body>
 <h1>Links</h1>
 <?php echo $this->getChildHtml('top.links'); ?>
</body>
```

When you reload the page, you'll notice that your `<h1>Links</h1>` title is rendering, but nothing is rendering for `top.links`. That's because we didn't add it to `local.xml`. The `getChildHtml` method can only include Blocks that are specified as sub-Blocks in the Layout. This allows Magento to only instantiate the Blocks it needs, and also allows you to set different Templates for Blocks based on context.

Let's add the `top.links` Block to our `local.xml`

```
<helloworld_index_index>
 <block type="page/html" name="root" output="toHtml" template="magentotutorial/helloworld/simple_page.phtml">
 <block type="page/template.links" name="top.links"/>
 <block type="customer/form_register" name="customer_form_register" template="customer/form/register.phtml"/>
 </block>
</helloworld_index_index>
```

Clear your cache and reload the page. You should now see the `top.links` module.

## Time for action

There is one more important concept to cover before we wrap up this lesson, and that is the `<action />` tag. Using the `<action />` tag enables us to call public PHP methods of the block classes. So instead of changing the template of the root block by replacing the block instance with our own, we can use a call to `setTemplate` instead.

```
<layout version="0.1.0">
<helloworld_index_index>
 <reference name="root">
 <action method="setTemplate">
 <template>magentotutorial/helloworld/simple_page.phtml</templat
ate>
 </action>
 <block type="page/template_links" name="top.links"/>
 <block type="customer/form_register" name="customer_form_register
" template="customer/form/register.phtml"/>
 </reference>
</helloworld_index_index>
</layout>
```

This layout XML will first set the template property of the root block, and then will add the two blocks we use as child blocks. Once we clear the cache, the result should look just as before. The benefit of using the `<action />` is the same block instance is used that was created earlier, and all other parent/child associations still exist. For that reason this is a more upgrade proof way of implementing our changes.

All arguments to the action's method need to be wrapped in an individual child node of the `<action />` tag. The name of that node doesn't matter, only the order of the nodes. We could have written the action node from the previous example as follows with the same effect.

```
<action method="setTemplate">
 <some_new_template>magentotutorial/helloworld/simple_page.phtml</some_new
_template>
</action>
```

This is just to illustrate that the action's argument node names are arbitrary.

## Wrapup

That covers Layout fundamentals. We have covered the tags `<block />`, `<reference />`, `<update />` and `<action />`, and also layout update handles like `<default />` and `<cms_index_index />`. These make up most of the layout configuration used in Magento. If you found it somewhat daunting, don't worry, you'll rarely need to work with layouts on such a fundamental level. Magento provides a number of pre-built layouts which can be modified and skinned to meet the needs of your store. Understanding how the entire Layout system works can be a great help when you're trouble shooting Layout issues, or adding new functionality to an existing Magento system.

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento for Developers: Part 5—Magento Models and ORM Basics

by *Alan Storm*, updated for Magento 1.12

[Edit this page on GitHub](#) 

### Other articles in this series:

- [Part 1—Introduction to Magento](#)
- [Part 2—The Magento Config](#)
- [Part 3—Magento Controller Dispatch](#)
- [Part 4—Magento Layouts, Blocks and Templates](#)
- [Part 5—Magento Models and ORM Basics](#)
- [Part 6—Magento Setup Resources](#)
- [Part 7—Advanced ORM: Entity Attribute Value](#)
- [Part 8—Varien Data Collections](#)

The implementation of a "Models Tier" is a huge part of any MVC framework. It represents the data of your application, and most applications are useless without data. Magento Models play an even bigger role, as they typically contain the "Business Logic" that's often relegated to the Controller or Helper methods in other PHP MVC frameworks.

## Traditional PHP MVC Models

If the definition of MVC is somewhat fuzzy, the definition of a Model is even fuzzier. Prior to the wide adoption of the MVC pattern by PHP developers, data access was usually raw SQL statements and/or an SQL abstraction layer. Developers would write queries and not think too much about what objects they were modeling.

In this day and age, raw SQL is mostly frowned upon, but many PHP frameworks are still SQL centric. Models will be objects that provide some layer of abstraction, but behind the scenes developers are still writing SQL and/or calling SQL like abstraction methods to read and write-down their data.

Other frameworks eschew SQL and take the Object Relational Mapping (ORM) approach. Here, a developer is dealing strictly with Objects. Properties are set, and when a save method is called on the Object, the data is automatically written to the database. Some ORMs will attempt to divine object properties from the database, others require the user to specify them in some way, (usually in an abstract data language such as YAML). One of the most famous and popular implementations of this approach is ActiveRecord.

This definition of ORM should suffice for now, but like everything Computer Science these days, the strict definition of ORM has blurred over the years. It's beyond the scope of this article to settle that dispute, but suffice it say we're generalizing a bit.

## Magento Models

It should be no surprise that Magento takes the ORM approach. While the Zend Framework SQL abstractions are available, most of your data access will be via the built in Magento Models, and Models you build yourself. It should also come as no surprise that Magento has a highly flexible, highly abstract, concept of what a Model is.

### Anatomy of a Magento Model

Most Magento Models can be categorized in one of two ways. There's a basic, ActiveRecord-like/one-object-one-table Model, and there's also an Entity Attribute Value (EAV) Model. Each Model also gets a Model Collection. Collections are PHP objects used to hold a number of individual Magento Model instances. The Magento team has implemented the PHP Standard Library interfaces of IteratorAggregate and Countable to allow each Model type to have its own collection type. If you're not familiar with the PHP Standard Library, think of Model Collections as arrays that also have methods attached.

Magento Models don't contain any code for connecting to the database. Instead, each Model uses a `modelResource` class, that is used to communicate with the database server (via one read and one write adapter object). By decoupling the logical Model and the code that talks to the database, it's theoretically possible to write new resource classes for a different database schemas and platforms while keeping your Models themselves untouched.

### Enable developer mode

Something you should do in development—but *never* in production—is to enable Magento's `developer mode` which, among other things, displays exceptions in your browser. It's useful for debugging your code.

Enable developer mode in any of the following ways:

- [developer mode](#)
- Edit the `.htaccess` in the Magento root directory file to add SetEnv  
`MAGE_IS_DEVELOPER_MODE "true"`

### Creating a Basic Model

To begin, we're going to create a basic Magento Model. PHP MVC tradition insists we model a weblog post. The steps we'll need to take are

1. Create a new "Weblog" module
2. Create a database table for our Model
3. Add Model information to the config for a Model named Blogpost
4. Add Model Resource information to the config for the Blogpost Model
5. Add a Read Adapter to the config for the Blogpost Model
6. Add a Write Adapter to the config for the Blogpost Model
7. Add a PHP class file for the Blogpost Model
8. Add a PHP class file for the Blogpost Resource Model
9. Instantiate the Model

### Create a Weblog Module

You should be an old hat at creating empty modules at this point, so we'll skip the details and assume you can create an empty module named Weblog. After you've done that, we'll setup a route for an index Action Controller with an action named "testModel". As always, the following examples assume a Package Name of "Magentotutorial".

In `'Magentotutorial/Weblog/etc/config.xml'`, setup the following route

```
<frontend>
 <routers>
 <weblog>
 <use>standard</use>
 <args>
 <module>Magentotutorial_Weblog</module>
 <frontName>weblog</frontName>
 </args>
 </weblog>
 </routers>
```

```
</frontend>
```

And then add the following Action Controller in

```
class Magentotutorial_Weblog_IndexController extends Mage_Core_Controller_Front_Action {
 public function testModelAction() {
 echo 'Setup!';
 }
}
```

at Magentotutorial/Weblog/controllers/IndexController.php. Clear your Magento cache and load the following URL to ensure everything's been setup correctly.

```
http://example.com/weblog/index/testModel
```

You should see the word "Setup" on a white background.

## Creating the Database Table

Magento has a system for automatically creating and changing your database schemas, but for the time being we'll just manually create a table for our Model.

Using the command-line or your favorite MySQL GUI application, create a table with the following schema

```
CREATE TABLE `blog_posts` (
 `blogpost_id` int(11) NOT NULL auto_increment,
 `title` text,
 `post` text,
 `date` datetime default NULL,
 `timestamp` timestamp NOT NULL default CURRENT_TIMESTAMP,
 PRIMARY KEY (`blogpost_id`)
)
```

And then populate it with some data

```
INSERT INTO `blog_posts` VALUES (1,'My New Title','This is a blog post','2010-07-01 00:00:00','2010-07-02 23:12:30');
```

## The Global Config and Creating The Model

There are three individual things we need to setup for a Model in our config.

1. Enabling Models in our Module
2. Enabling Model Resources in our Module
3. Add an "entity" table configuration to our Model Resource.

When you instantiate a Model in Magento, you make a call like this

```
$model = Mage::getModel('weblog/blogpost');
```

The first part of the URI you pass into get Model is the **Model Group Name**. Because it is a good idea to follow conventions, this should be the (lowercase) name of your module, or to be safeguarded against conflicts use the packagename and modulename (also in lowercase). The second part of the URI is the lowercase version of your Model name.

So, let's add the following XML to our module's 'config.xml'.

```
<global>
 <!-- ... -->
 <models>
 <weblog>
 <class>Magentotutorial_Weblog_Model</class>
 <!--
 need to create our own resource, can't just
 use core_resource
 -->
 <resourceModel>weblog_resource</resourceModel>
 </weblog>
 </models>
 <!-- ... -->
</global>
```

The outer `<weblog />` tag is your Group Name, which should match your module name. `<class />` is the BASE name all Models in the weblog group will have, also called **Class Prefix**. The `<resourceModel />` tag indicates which Resource Model that weblog group Models should use. We talk more about this later on in this page. For now, remember your Group Name and the literal string "resource".

So, we're not done yet, but let's see what happens if we clear our Magento cache and attempt to instantiate a blogpost Model. In your `'testModelAction'` method, use the following code

```
public function testModelAction() {
 $blogpost = Mage::getModel('weblog/blogpost');
 echo get_class($blogpost);
}
```

and reload your page. You should see an error like the following:

```
include(Magentotutorial/Weblog/Model/Blogpost.php) [function.include]: failed
to open stream: No such file or directory
```

By attempting to retrieve a `weblog/blogpost` Model, you told Magento to instantiate a class with the name

```
Magentotutorial_Weblog_Model_Blogpost
```

Magento is trying to `__autoload` include this Model, but can't find the file. Let's create it! Create the following class at the following location

```
File: app/code/local/Magentotutorial/Weblog/Model/Blogpost.php
```

```
class Magentotutorial_Weblog_Model_Blogpost extends Mage_Core_Model_Abstract
{
 protected function _construct()
 {
 $this->_init('weblog/blogpost');
 }
}
```

Reload your page, and the exception should be replaced with the name of your class.

All basic Models that interact with the database should extend the `Mage_Core_Model_Abstract` class. This abstract class forces you to implement a single method named `_construct` (**NOTE:** this is not PHP's constructor `__construct`). This method should call the class's `_init` method with the same identifying URI you'll be using in the `Mage::getModel` method call.

## The Global Config and Resources

So, we've setup our Model. Next, we need to setup our Model Resource. Model Resources contain the code that actually talks to our database. In the last section, we included the following in our config.

```
<resourceModel>weblog_resource</resourceModel>
```

The value in `<resourceModel />` will be used to instantiate a Model Resource class. Although you'll never need to call it yourself, when any Model in the weblog group needs to talk to the database, Magento will make the following method call to get the Model resource

```
Mage::getResourceModel('weblog/blogpost');
```

Again, `weblog` is the Group Name, and `blogpost` is the Model. The `Mage::getResourceModel` method will use the `weblog/blogpost` URI to inspect the global config and pull out the value in `<resourceModel>` (in this case, `weblog_resource`). Then, a model class will be instantiated with the following URI

```
weblog_resource/blogpost
```

So, if you followed that all the way, what this means is, **resource models are configured in the same section of the XML config as normal Models**. This can be confusing to newcomers and old-hands alike.

So, with that in mind, let's configure our resource. In our `<models>` section add

```
<global>
 <!-- ... -->
```

```

<models>
 <!-- ... -->
 <weblog_resource>
 <class>Magentotutorial_Weblog_Model_Resource</class>
 </weblog_resource>
</models>
</global>

```

You're adding the `<weblog_resource />` tag, which is the value of the `<resourceModel />` tag you just setup. The value of `<class />` is the base name that all your resource models will have, and should be named with the following format

Packagename\_Modulename\_Model\_Resource

So, we have a configured resource, let's try loading up some Model data. Change your action to look like the following

```

public function testModelAction() {
 $params = $this->getRequest()->getParams();
 $blogpost = Mage::getModel('weblog/blogpost');
 echo("Loading the blogpost with an ID of ".$params['id']);
 $blogpost->load($params['id']);
 $data = $blogpost->getData();
 var_dump($data);
}

```

And then load the following URL in your browser (after clearing your Magento cache)

<http://example.com/weblog/index/testModel/id/1>

You should see an exception something like the following

```

Warning: include(Magentotutorial/Weblog/Model/Resource/Blogpost.php) [function
n.include]: failed to open stream: No such file

```

As you've likely intuited, we need to add a resource class for our Model. **Every** Model has its own resource class. Add the following class at the following location

File: app/code/local/Magentotutorial/Weblog/Model/Resource/Blogpost.php

```

class Magentotutorial_Weblog_Model_Resource_Blogpost extends Mage_Core_Model_
Resource_Db_Abstract{
 protected function _construct()
 {
 $this->_init('weblog/blogpost', 'blogpost_id');
 }
}

```

Again, the first parameter of the init method is the URL used to identify the **Model**. The second parameter is the database field that uniquely identifies any particular column. In most cases, this should be the primary key. Clear your cache, reload, and you should see

Can't retrieve entity config: weblog/blogpost

Another exception! When we use the Model URI `weblog/blogpost`, we're telling Magento we want the Model Group `weblog`, and the `blogpost Entity`. In the context of simple Models that extend `Mage_Core_Model_Resource_Db_Abstract`, an entity corresponds to a table. In this case, the table named `blog_post` that we created above. Let's add that entity to our XML config.

```

<models>
 <!-- ... -->
 <weblog_resource>
 <class>Magentotutorial_Weblog_Model_Resource</class>
 <entities>
 <blogpost>
 <table>blog_posts</table>
 </blogpost>
 </entities>
 </weblog_resource>
</models>

```

We've added a new `<entities />` section to the resource Model section of our config. This, in turn,

has a section named after our entity (<blogpost />) that specifies the name of the database table we want to use for this Model.

Clear your Magento cache, cross your fingers, reload the page and ...

```
Loading the blogpost with an ID of 1

array
'blogpost_id' => string '1' (length=1)
'title' => string 'My New Title' (length=12)
'post' => string 'This is a blog post' (length=19)
'date' => string '2009-07-01 00:00:00' (length=19)
'timestamp' => string '2009-07-02 16:12:30' (length=19)
```

Eureka! We've managed to extract our data and, more importantly, completely configure a Magento Model.

## Basic Model Operations

All Magento Models inherit from the Varien\_Object class. This class is part of the Magento system library and **not** part of any Magento core module. You can find this object at

```
lib/Varien/Object.php
```

Magento Models store their data in a protected \_data property. The Varien\_Object class gives us several methods we can use to extract this data. You've already seen `getData`, which will return an array of key/value pairs. This method can also be passed a string key to get a specific field.

```
$model->getData();
$model->getData('title');
```

There's also a `getOrigData` method, which will return the Model data as it was when the object was initially populated, (working with the protected `_origData` method).

```
$model->getOrigData();
$model->getOrigData('title');
```

The Varien\_Object also implements some special methods via PHP's magic `__call` method. You can get, set, unset, or check for the existence of any property using a method that begins with the word get, set, unset or has and is followed by the camel cased name of a property.

```
$model->getBlogpostId();
$model->setBlogpostId(25);
$model->unsBlogpostId();
if($model->hasBlogpostId()){...}
```

For this reason, you'll want to name all your database columns with lower case characters and use underscores to separate characters.

## CRUD, the Magento Way

Magento Models support the basic Create, Read, Update, and Delete functionality of CRUD with `load`, `save`, and `delete` methods. You've already seen the `load` method in action. When passed a single parameter, the `load` method will return a record whose id field (set in the Model's resource) matches the passed in value.

```
$blogpost->load(1);
```

The `save` method will allow you to both `INSERT` a new Model into the database, or `UPDATE` an existing one. Add the following method to your Controller

```
public function createNewPostAction() {
 $blogpost = Mage::getModel('weblog/blogpost');
 $blogpost->setTitle('Code Post!');
 $blogpost->setPost('This post was created from code!');
 $blogpost->save();
 echo 'post with ID ' . $blogpost->getId() . ' created';
}
```

and then execute your Controller Action by loading the following URL

```
http://example.com/weblog/index/createNewPost
```

You should now see an additional saved post in your database table. Next, try the following to edit your post

```
public function editFirstPostAction() {
 $blogpost = Mage::getModel('weblog/blogpost');
 $blogpost->load(1);
 $blogpost->setTitle("The First post!");
 $blogpost->save();
 echo 'post edited';
}
```

And finally, you can delete your post using very similar syntax.

```
public function deleteFirstPostAction() {
 $blogpost = Mage::getModel('weblog/blogpost');
 $blogpost->load(1);
 $blogpost->delete();
 echo 'post removed';
}
```

## Model Collections

So, having a single Model is useful, but sometimes we want to grab list of Models. Rather than returning a simple array of Models, each Magento Model type has a unique collection object associated with it. These objects implement the PHP IteratorAggregate and Countable interfaces, which means they can be passed to the count function, and used in for each constructs.

We'll cover Collections in full in a later article, but for now let's look at basic setup and usage. Add the following action method to your Controller, and load it in your browser.

```
public function showAllBlogPostsAction() {
 $posts = Mage::getModel('weblog/blogpost')->getCollection();
 foreach($posts as $blogpost){
 echo '

' . $blogpost->getTitle() . '

';
 echo nl2br($blogpost->getPost());
 }
}
```

Load the action URL,

```
http://example.com/weblog/index/showAllBlogPosts
```

and you should see a (by now) familiar exception.

```
Warning: include(Magentotutorial/Weblog/Model/Resource/Blogpost/Collection.php) [function.include]: failed to open stream
```

You're not surprised, are you? We need to add a PHP class file that defines our Blogpost collection. Every Model has a protected property named `_resourceCollectionName` that contains a URI that's used to identify our collection.

```
protected '_resourceCollectionName' => string 'weblog/blogpost_collection'
```

By default, this is the same URI that's used to identify our Resource Model, with the string `_collection` appended to the end. Magento considers Collections part of the Resource, so this URI is converted into the class name.

```
Magentotutorial_Weblog_Model_Resource_Blogpost_Collection
```

Add the following PHP class at the following location

```
File: app/code/local/Magentotutorial/Weblog/Model/Resource/Blogpost/Collection.php
```

```
class Magentotutorial_Weblog_Model_Resource_Blogpost_Collection extends Mage_Core_Model_Resource_Db_Collection_Abstract {
 protected function _construct()
 {
 $this->_init('weblog/blogpost');
 }
}
```

Just as with our other classes, we need to init our Collection with the Model URI. (weblog/blogpost). Rerun your Controller Action, and you should see your post information.

## Wrapup

Congratulations, you've created and configured your first Magento Model. In a later article we'll take a look at Magento's Entity Attribute Value Models (EAV), which expand on what we've learned here.

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento for Developers: Part 6—Magento Setup Resources

by [Alan Storm](#), updated for Magento 1.12

[Edit this page on GitHub](#) 

### Other articles in this series:

- [Part 1—Introduction to Magento](#)
- [Part 2—The Magento Config](#)
- [Part 3—Magento Controller Dispatch](#)
- [Part 4—Magento Layouts, Blocks and Templates](#)
- [Part 5—Magento Models and ORM Basics](#)
- [Part 6—Magento Setup Resources](#)
- [Part 7—Advanced ORM: Entity Attribute Value](#)
- [Part 8—Varien Data Collections](#)

On any fast paced software development project, the task of keeping the development and production databases in sync become a sticky wicket. Magento offers a system to create versioned resource migration scripts that can help your team deal with this often contentious part of the development process.

In the [ORM article](#) we created a model for a weblog post. At the time, we ran our `CREATE TABLE` statements directly against the database. This time, we'll create a Setup Resource for our module that will create the table for us. We'll also create an upgrade script for our module that will update an already installed module. The steps we'll need to take are

1. Add the Setup Resource to our config
2. Create our resource class file
3. Create our installer script
4. Create our upgrade script

## Adding the Setup Resource

So, let's continue with the weblog module [we created last time](#). In our `<global />` section, add the following

```
<global>
 <!-- ... -->
 <resources>
 <weblog_setup>
 <setup>
```

```

<module>Magentotutorial_Weblog</module>
<class>Magentotutorial_Weblog_Model_Resource_Setup</class>
</setup>
</weblog_setup>
</resources>
<!-- ... -->
</global>

```

The `<weblog_setup>` tag will be used to uniquely identify this Setup Resource. It's encouraged, but not necessary, that you use the `modelname_setup` naming convention. The `<module>Magentotutorial_Weblog</module>` tag block should contain the `Packagename_ModuleName` of your module. Finally, `<class>Magentotutorial_Weblog_Model_Resource_Setup</class>` should contain the name of the class we'll be creating for our Setup Resource. For basic setup scripts it's not necessary to create a custom class, but by doing it now you'll give yourself more flexibility down the line.

After adding the above section to your config, clear your Magento cache and try to load any page of your Magento site. You'll see an exception something like

```
Fatal error: Class 'Magentotutorial_Weblog_Model_Resource_Setup' not found in
```

Magento just tried to instantiate the class you specified in your config, but couldn't find it. You'll want to create the following file, with the following contents.

File: app/code/local/Magentotutorial/Weblog/Model/Resource/Setup.php

```
class Magentotutorial_Weblog_Model_Resource_Setup extends Mage_Core_Model_Resource_Setup {
}
```

Now, reload any page of your Magento site. The exception should be gone, and your page should load as expected.

## Creating our Installer Script

Next, we'll want to create our installer script. This is the script that will contain any `CREATE TABLE` or other SQL code that needs to be run to initialize our module.

First, take a look at your config.xml file

```

<modules>
 <Magentotutorial_Weblog>
 <version>0.1.0</version>
 </Magentotutorial_Weblog>
</modules>

```

This section is required in all config.xml files, and identifies the module as well as the its version number. Your installer script's name will be based on this version number. The following assumes the current version of your module is 0.1.0.

Create the following file at the following location

File: app/code/local/Magentotutorial/Weblog/sql/weblog\_setup/mysql4-install-0.1.0.php

```
echo 'Running This Upgrade: '.get_class($this)."\n
 \n";
die("Exit for now");
```

The `weblog_setup` portion of the path should match the tag you created in your config.xml file (`<weblog_setup>`). The `0.1.0` portion of the filename should match the starting version of your module. Clear your Magento cache and reload any page in your Magento site and you should see something like

```
Running This Upgrade: Magentotutorial_Weblog_Model_Resource_Setup
Exit for now
...
```

Which means your update script ran. Eventually we'll put our SQL update scripts here, but for now we're going to concentrate on the setup mechanism itself. Remove the `die` statement from your script so it looks like the following

```
echo 'Running This Upgrade: '.get_class($this)."\n
 \n";
```

Reload your page. You should see your upgrade message displayed at the top of the page. Reload again, and your page should be displayed as normal.

## Resource Versions

Magento's Setup Resources allow you to simply drop your install scripts (and upgrade scripts, which we'll get to in a bit) onto the server, and have the system automatically run them. This allows you to have all your database migrations scripts stored in the system in a consistent format.

Using your favorite database client, take a look at the core\_setup table

```
mysql> select * from core_resource;
+-----+-----+-----+
| code | version | data_version |
+-----+-----+-----+
| adminnotification_setup | 1.6.0.0 | 1.6.0.0 |
| admin_setup | 1.6.1.0 | 1.6.1.0 |
| api2_setup | 1.0.0.0 | 1.0.0.0 |
| api_setup | 1.6.0.0 | 1.6.0.0 |
| backup_setup | 1.6.0.0 | 1.6.0.0 |
| bundle_setup | 1.6.0.0.1 | 1.6.0.0.1 |
| captcha_setup | 1.7.0.0.0 | 1.7.0.0.0 |
| catalogindex_setup | 1.6.0.0 | 1.6.0.0 |
| cataloginventory_setup | 1.6.0.0.2 | 1.6.0.0.2 |
| catalogrule_setup | 1.6.0.3 | 1.6.0.3 |
| catalogsearch_setup | 1.6.0.0 | 1.6.0.0 |
| catalog_setup | 1.6.0.0.14 | 1.6.0.0.14 |
| checkout_setup | 1.6.0.0 | 1.6.0.0 |
| cms_setup | 1.6.0.0.1 | 1.6.0.0.1 |
| compiler_setup | 1.6.0.0 | 1.6.0.0 |
| contacts_setup | 1.6.0.0 | 1.6.0.0 |
| core_setup | 1.6.0.2 | 1.6.0.2 |
| cron_setup | 1.6.0.0 | 1.6.0.0 |
| customer_setup | 1.6.2.0.1 | 1.6.2.0.1 |
| dataflow_setup | 1.6.0.0 | 1.6.0.0 |
| directory_setup | 1.6.0.1 | 1.6.0.1 |
| downloadable_setup | 1.6.0.0.2 | 1.6.0.0.2 |
| eav_setup | 1.6.0.0 | 1.6.0.0 |
| giftmessage_setup | 1.6.0.0 | 1.6.0.0 |
| googleanalytics_setup | 0.1.0 | 0.1.0 |
| googlecheckout_setup | 1.6.0.1 | 1.6.0.1 |
| importexport_setup | 1.6.0.2 | 1.6.0.2 |
| index_setup | 1.6.0.0 | 1.6.0.0 |
| log_setup | 1.6.0.0 | 1.6.0.0 |
| moneybookers_setup | 1.6.0.0 | 1.6.0.0 |
| newsletter_setup | 1.6.0.1 | 1.6.0.1 |
| oauth_setup | 1.0.0.0 | 1.0.0.0 |
| paygate_setup | 1.6.0.0 | 1.6.0.0 |
| payment_setup | 1.6.0.0 | 1.6.0.0 |
| paypaluk_setup | 1.6.0.0 | 1.6.0.0 |
| paypal_setup | 1.6.0.2 | 1.6.0.2 |
| persistent_setup | 1.0.0.0 | 1.0.0.0 |
| poll_setup | 1.6.0.0 | 1.6.0.0 |
| productalert_setup | 1.6.0.0 | 1.6.0.0 |
| rating_setup | 1.6.0.0 | 1.6.0.0 |
| reports_setup | 1.6.0.0.1 | 1.6.0.0.1 |
| review_setup | 1.6.0.0 | 1.6.0.0 |
| salesrule_setup | 1.6.0.3 | 1.6.0.3 |
| sales_setup | 1.6.0.7 | 1.6.0.7 |
| sendfriend_setup | 1.6.0.0 | 1.6.0.0 |
| shipping_setup | 1.6.0.0 | 1.6.0.0 |
| sitemap_setup | 1.6.0.0 | 1.6.0.0 |
| tag_setup | 1.6.0.0 | 1.6.0.0 |
| tax_setup | 1.6.0.3 | 1.6.0.3 |
| usa_setup | 1.6.0.1 | 1.6.0.1 |
| weblog_setup | 0.1.0 | 0.1.0 |
| weeSetup | 1.6.0.0 | 1.6.0.0 |
| widget_setup | 1.6.0.0 | 1.6.0.0 |
| wishlist_setup | 1.6.0.0 | 1.6.0.0 |
| xmlconnect_setup | 1.6.0.0 | 1.6.0.0 |
+-----+-----+-----+
55 rows in set (0.00 sec)
```

This table contains a list of all the installed modules, along with the installed version number. You can see our module near the end

```
| weblog_setup | 0.1.0 | 0.1.0 |
```

This is how Magento knows not to re-run your script on the second, and on all successive, page loads. The `weblog_setup` is already installed, so it won't be updated. If you want to re-run your installer script (useful when you're developing), just delete the row for your module from this table. Let's do that now, and actually add the SQL to create our table. So first, run the following SQL.

```
DELETE from core_resource where code = 'weblog_setup';
```

We'll also want to drop the table we manually created in the [ORM article](#).

```
DROP TABLE blog_posts;
```

Then, add the following code to your setup script.

```
$installer = $this;
$installer->startSetup();
$installer->run("
 CREATE TABLE `{$installer->getTable('weblog/blogpost')}` (
 `blogpost_id` int(11) NOT NULL auto_increment,
 `title` text,
 `post` text,
 `date` datetime default NULL,
 `timestamp` timestamp NOT NULL default CURRENT_TIMESTAMP,
 PRIMARY KEY (`blogpost_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

 INSERT INTO `{$installer->getTable('weblog/blogpost')}`
 VALUES (1,'My New Title','This is a blog post','2
009-07-01 00:00:00','2009-07-02 23:12:30');
");
$installer->endSetup();
```

Clear your Magento cache and reload any page in the system. You should have a new blog\_posts table with a single row.

## Anatomy of a Setup Script

So, let's go over the script line-by-line. First, there's this (or is that \$this?)

```
$installer = $this;
```

Each installer script is run from the context of a Setup Resource class, the class you created above. That means any reference to \$this from within the script will be a reference to an object instantiated from this class. While not necessary, most setup scripts in the core modules will alias \$this to a variable called installer, which is what we've done here. While not necessary, it is the convention and it's always best to follow the convention unless you have a good reason for breaking it.

Next, you'll see our queries are bookended by the following two method calls.

```
$installer->startSetup();
//...
$installer->endSetup();
```

If you take a look at the Mage\_Core\_Model\_Resource\_Setup class in app/code/core/Mage/Core/Model/Resource/Setup.php (which your setup class inherits from) you can see that these methods do some basic SQL setup

```
public function startSetup()
{
 $this->getConnection()->startSetup()
 return $this;
}

public function endSetup()
{
 $this->getConnection()->endSetup();
 return $this;
}
```

You can look into Varien\_Db\_Adapter\_Pdo\_Mysql in lib/varien/Db/Adapter/Pdo/Mysql.php to find the real SQL setup executed for MySQL connections in the startSetup() and endSetup() methods.

Finally, there's the call to the run method

```
$installer->run(...);
```

which accepts a string containing the SQL needed to setup your database table(s). You may specify any number of queries, separated by a semi-colon. You also probably noticed the following

```
$installer->getTable('weblog/blogpost')
```

The getTable method allows you to pass in a Magento Model URI and get its table name. While not necessary, using this method ensures that your script will continue to run, even if someone changes the name of their table in the config file. The Mage\_Core\_Model\_Resource\_Setup class contains many useful helper methods like this. The best way to become familiar with everything that's possible is to study the installer scripts used by the core Magento modules.

## RDBMS Agnostic Scripts

Since version 1.6, Magento (in theory) supports more database backends than only MySQL. Since our setup script contains raw SQL statements, it may not run correctly on a different database system, say MSSQL. For that reason the setup script name is prefix with the string `mysql4-`

In order to make setup scripts cross-database compatible, Magento offers a DDL (Data Definition Language) Table object. Here is an alternative version of our setup script that would run on any supported RDBMS.

File: app/code/local/Magentotutorial/Weblog/sql/weblog\_setup/mysql4-install-0.1.0.php

```
$installer = $this;
$installer->startSetup();
$table = $installer->getConnection()->newTable($installer->getTable('weblog/blogpost'))
->addColumn('blogpost_id', Varien_Db_Ddl_Table::TYPE_INTEGER, null, array(
 'unsigned' => true,
 'nullable' => false,
 'primary' => true,
 'identity' => true,
), 'Blogpost ID')
->addColumn('title', Varien_Db_Ddl_Table::TYPE_TEXT, null, array(
 'nullable' => false,
), 'Blogpost Title')
->addColumn('post', Varien_Db_Ddl_Table::TYPE_TEXT, null, array(
 'nullable' => true,
), 'Blogpost Body')
->addColumn('date', Varien_Db_Ddl_Table::TYPE_DATETIME, null, array(
), 'Blogpost Date')
->addColumn('timestamp', Varien_Db_Ddl_Table::TYPE_TIMESTAMP, null, array(
), 'Timestamp')
->setComment('Magentotutorial weblog/blogpost entity table');
$installer->getConnection()->createTable($table);

$installer->endSetup();
```

As you can see, there is no raw SQL in this version of the setup script. So which version should you use? If you want your Modules to run on any RDBMS backend, use the new DDL style upgrade scripts. If you are concerned about backward compatibility, use the raw SQL flavor, that is still supported by Magento 1.6 and 1.7 (and probably will be supported by any 1.x Magento release).

## Module Upgrades

So, that's how you create a script that will setup your initial database tables, but what if you want to alter the structure of an existing module? Magento's Setup Resources support a simple versioning scheme that will let you automatically run scripts to **upgrade** your modules.

Once Magento runs an *installer* script for a module, it will **never run another installer for that module again** (short of manually deleting the reference in the `core_resource` table). Instead, you'll need to create an upgrade script. Upgrade scripts are very similar to installer scripts, with a few key differences.

To get started, we'll create a script at the following location, with the following contents

File: app/code/local/Magentotutorial/Weblog/sql/weblog\_setup/upgrade-0.1.0-0.2.0.php:

```
echo 'Testing our upgrade script (upgrade-0.1.0-0.2.0.php) and halting execution to avoid updating the system
version number
';
die();
```

Upgrade scripts are placed in the same folder as your installer script, but named slightly differently. First, and most obviously, the file name contains the word `upgrade`. Secondly, you'll notice there are **two** version numbers, separated by a `"."`. The first (`0.1.0`) is the module version that we're upgrading **from**. The second (`0.2.0`) is the module version we're upgrading **to**.

If we cleared our Magento cache and reloaded a page, our script wouldn't run. We need to update the version number in our module's config.xml file to trigger the upgrade

```
<modules>
 <Magentotutorial_Weblog>
 <version>0.2.0</version>
 </Magentotutorial_Weblog>
</modules>
```

With the new version number in place, we'll need to clear our Magento cache and load any page in our Magento site. You should now see output from your upgrade script.

By the way, we also could have names our upgrade script `mysql4-upgrade-0.1.0-0.2.0.php`. This would indicate our upgrade would contain MySQL specific SQL.

Before we continue and actually implement the upgrade script, there's one important piece of behavior you'll want to be aware of. Create another upgrade file at the following location with the following contents.

File: app/code/local/Magentotutorial/Weblog/sql/weblog\_setup/upgrade-0.1.0-0.1.5.php:

```
echo 'Testing our upgrade script (upgrade-0.1.0-0.1.5.php) and NOT halting execution
';
```

If you reload a page, you'll notice you see BOTH messages. When Magento notices the version number of a module has changed, it will run through **all** the setup scripts needed to bring that version up to date. Although we never really created a version 0.1.5 of the Weblog module, Magento sees the upgrade script, and will attempt to run it. Scripts will be run in order from lowest to highest. If you take a peek at the `core_resource` table,

```
mysql> select * from core_resource where code = 'weblog_setup';
+-----+-----+-----+
| code | version | data_version |
+-----+-----+-----+
| weblog_setup | 0.1.5 | 0.1.5 |
+-----+-----+
1 row in set (0.00 sec)
```

you'll notice Magento considers the version number to be 1.5. That's because we completed executing the 1.0 to 1.5 upgrade, but did not complete execution of the 1.0 to 2.0 upgrade.

So, with all that out of the way, writing our actual upgrade script is identical to writing an installer script. Let's change the `0.1.0-0.2.0` script to read

```
$installer = $this;
$installer->startSetup();
$installer->getConnection()
 ->changeColumn($installer->getTable('weblog/blogpost'), 'post', 'post', array(
 'type' => Varien_Db_Ddl_Table::TYPE_TEXT,
 'nullable' => false,
 'comment' => 'Blogpost Body'
)
);
$installer->endSetup();
die("You'll see why this is here in a second");
```

Try refreshing a page in your Magento site and ... nothing. The upgrade script didn't run. The post field in our table still allows null values, and more importantly, the call to `die()` did not halt execution. Here's what happened

1. The `weblog_setup` resource was at version 0.1.0
2. We upgraded our module to version 0.2.0
3. Magento saw the upgraded module, and saw there were two upgrade scripts to run; 0.1.0 to 0.1.5 and 0.1.0 to 0.2.0
4. Magento queued up both scripts to run
5. Magento ran the 0.1.0 to 0.1.5 script
6. The `weblog_setup` resource is now at version 0.1.5
7. Magento ran the 0.1.0 to 0.2.0 script, execution was halted
8. On the next page load, Magento saw `weblog_setup` at version 0.1.5 and did not see any upgrade scripts to run since both scripts indicated they should be run `from 0.1.0`

The correct way to achieve what we wanted would have been to name our scripts as follows

```
upgrade-0.1.0-0.1.5.php #This goes from 0.1.0 to 0.1.5
upgrade-0.1.5-0.2.0.php #This goes 0.1.5 to 0.2.0
```

Magento is smart enough to run both scripts on a single page load. You can go back in time and give this a try by updating the `core_resource` table

```
UPDATE core_resource SET version = '0.1.0', data_version = '0.1.0' WHERE code = 'weblog_setup';
...
```

It's one of the odd quirks of the Magento system that the updates will run as previously configured. This means you'll want to be careful with multiple developers adding update scripts to the system. You'll either want a build-meister/deployment-manager type in charge of the upgrade scripts or (heaven forbid) developers will need to talk to one another.

## Wrap-up

You should now know the basics of how to use Magento Setup Resources to create versioned database migration scripts, as well as understand the scripts provided in the core modules. Beyond having a standard way for developers to write migration scripts, Setup Resources become much more important when creating and modifying Entity Attribute Value models.

22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento for Developers: Part 7—Advanced ORM: Entity Attribute Value

by Alan Storm, updated for Magento 1.12

[Edit this page on GitHub](#) 

### Other articles in this series:

- [Part 1—Introduction to Magento](#)
- [Part 2—The Magento Config](#)
- [Part 3—Magento Controller Dispatch](#)
- [Part 4—Magento Layouts, Blocks and Templates](#)
- [Part 5—Magento Models and ORM Basics](#)
- [Part 6—Magento Setup Resources](#)
- [Part 7—Advanced ORM: Entity Attribute Value](#)
- [Part 8—Varien Data Collections](#)

In the [first ORM article](#) we told you there were two kinds of Models in Magento. Regular, or "simple" Models, and Entity Attribute Value (or EAV) Models. We also told you this was a bit of a fib. Here's where we come clean.

ALL Magento Models interacting with the database inherit from the `Mage_Core_Model_Abstract` / `Varien_Object` chain. What makes something either a simple Model or an EAV Model is its **Model Resource**. While all resources extend the base `Mage_Core_Model_Resource_Abstract` class, simple Models have a resource that inherits from `Mage_Core_Model_Resource_Db_Abstract`, and EAV Models have a resource that inherits from `Mage_Eav_Model_Entity_Abstract`.

If you think about it, this makes sense. As the end-programmer-user of the system you want a set of methods you can use to talk to and manipulate your Models. You don't care what the backend storage looks like, you just want to get properties and invoke methods that trigger business rules.

## What is EAV

[Wikipedia defines EAV](#) as

Entity-Attribute-Value model (EAV), also known as object-attribute-value model and open schema is a data model that is used in circumstances where the number of attributes (properties, parameters) that can be used to describe a thing (an "entity" or "object") is potentially very vast, but the number that will actually apply to a given entity is relatively modest. In mathematics, this model is known as a sparse matrix.

Another metaphor that helps me wrap my head around it is "EAV brings some aspects of

normalization to the database **table schema**. In a traditional database, tables have a fixed number of columns

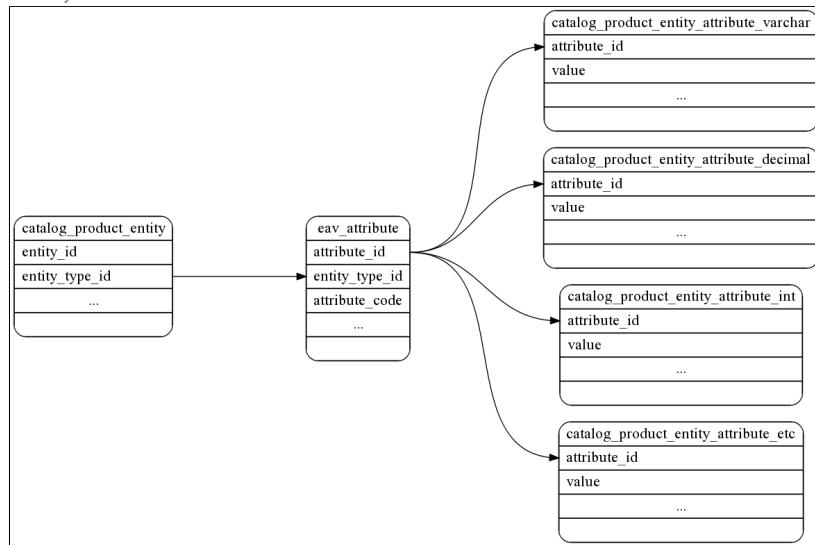
| products   |          |       |        |  |
|------------|----------|-------|--------|--|
| product_id | name     | price | etc... |  |
| 1          | Widget A | 11.34 | etc... |  |
| 2          | Dongle B | 6.34  | etc... |  |

Every product has a name, every product has a price, etc.

In an EAV Model, each "entity" (product) being modeled has a **different** set of attributes. EAV makes a lot of sense for a generic eCommerce solution. A store that sells laptops (which have a CPU speed, color, ram amount, etc) is going to have a different set of needs than a store that sells yarn (yarn has a color, but no CPU speed, etc.). Even within our hypothetical yarn store, some products will have length (balls of yarn), and others will have diameter (knitting needles).

There aren't many open source or commercial databases that use EAV by default. There are none that are available on a wide variety of web hosting platforms. Because of that, the Magento engineers have built an EAV system out of PHP objects that use MySQL as a data-store. In other words, they've built an EAV database system **on top of** a traditional relational database.

In practice this means any Model that uses an EAV resource has its attributes spread out over a number of MySQL tables.



The above diagram is a rough layout of the database tables Magento consults when it looks up an EAV record for the `catalog_product` entity. Each individual product has a row in `catalog_product_entity`. All the available attributes in the **entire** system (not just for products) are stored in `eav_attribute`, and the actual attribute values are stored in tables with names like `catalog_product_entity_varchar`, `catalog_product_entity_decimal`, `catalog_product_entity_int`, `catalog_product_entity/etc`...

Beyond the mental flexibility an EAV system gives you, there's also the practical benefit of avoiding `ALTER TABLE` statements. When you add a new attribute for your products, a new row is inserted into `eav_attribute`. In a traditional relational database/single-table system, you'd need to `ALTER` the actual database structure, which can be a time consuming/risky proposition for tables with large data-sets.

The downside is there's no one single simple SQL query you can use to get at all your product data. Several single SQL queries or one large join need to be made.

## Implementing EAV

That's EAV in a nutshell. The rest of this article is a run-through of what's needed to create a new EAV Model in Magento. It's the hairiest thing you'll read about Magento and it's something that 95%

of people working with the system will never need to do. However, understanding what it takes to build an EAV Model Resource will help you understand what's going on with the EAV Resources that Magento uses.

Because the EAV information is so dense, we're going to assume you're already very familiar with Magento's MVC and grouped class name features. We'll help you along the way, but training wheels are off.

## Weblog, EAV Style

We're going to create another Model for a weblog post, but this time using an EAV Resource. To start with, setup and create a new module which responds at the following URL

```
http://example.com/complexworld
```

If you're unsure how to do this, be sure you've mastered the concepts in the [previous tutorials](#).

Next, we'll create a new Model named Weblogeav. Remember, it's the **Resource** that's considered EAV. We design and configure our Model the exact same way, so let's configure a Model similar to one we created in the first ORM article.

```
<global>
 <!-- ... -->
 <models>
 <!-- ... -->
 <complexworld>
 <class>Magentotutorial_Complexworld_Model</class>
 <resourceModel>complexworld_resource</resourceModel>
 </complexworld>
 <!-- ... -->
 </models>
 <!-- ... -->
</global>
```

You'll notice so far there is no difference to setting up a regular Model and flat table resource Model.

We'll still need to let Magento know about this resource. Similar to basic Models, EAV Resources are configured in the same `<model/>` node with everything else.

```
<global>
 <!-- ... -->
 <models>
 <!-- ... -->
 <complexworld_resource>
 <class>Magentotutorial_Complexworld_Model_Resource</class>
 <entities>
 <eavblogpost>
 <table>eavblog_posts</table>
 </eavblogpost>
 </entities>
 </complexworld_resource>
 <!-- ... -->
 </models>
 <!-- ... -->
</global>
```

Again, so far this is setup similar to our regular Model Resource. We provide a `<class/>` that configures a PHP class, as well as an `<entities/>` section that will let Magento know the base table for an individual Model we want to create. The `<eavblogpost/>` tag is the name of the specific Model we want to create, and its inner `<table/>` tag specifies the base table this Model will use (more on this later).

## Where Does That File Go?

Until wide adoption of PHP 5.3 and namespaces, one of the trickier (and tedious) parts of Magento will remain remembering how `<classname/>`s relate to file paths, and then ensuring you create the correctly named directory structure and class files. After configuring any `<classname/>`s or URLs, you may find it useful to attempt to instantiate an instance of the class in a controller **without** first creating the class files. This way PHP will throw an exception telling me it can't find a file, along with the file location. Give the following a try in your Index Controller.

```
public function indexAction() {
 $weblog2 = Mage::getModel('complexworld/eavblogpost');
 $weblog2->load(1);
 var_dump($weblog2);
}
```

As predicted, a warning should be thrown

```
Warning: include(Magentotutorial/Complexworld/Model/Eavblogpost.php) [function.include]:
failed to open stream: No such file or directory in /Users/username/Sites/magento.dev/lib/Varien/Autoload.php on line 93
```

In addition to telling us the path where we'll need to define the new resource class this also serves as a configuration check. If we'd been warned with the following

```
Warning: include(Mage/Complexworld/Model/Eavblogpost.php) [function.include]:
failed to open stream: No such file or directory in /Users/username/Sites/magento.dev/lib/Varien/Autoload.php on line 93
```

we'd know our Model was misconfigured, as Magento was looking for the Model in `code/core/Mage` instead of `code/local/Magentotutorial`.

So, let's create our Model class

File: `app/code/local/Magentotutorial/Complexworld/Model/Eavblogpost.php`:

```
class Magentotutorial_Complexworld_Model_Eavblogpost extends Mage_Core_Model_Abstract {
 protected function _construct()
 {
 $this->_init('complexworld/eavblogpost');
 }
}
```

Remember, the Model itself is resource independent. A regular Model and an EAV Model both extend from the same class. It's the resource that makes them different.

Clear your Magento cache, reload your page, and you should see a **new** warning.

```
Warning: include(Magentotutorial/Complexworld/Model/Resource/Eavblogpost.php)
```

As expected, we need to create a class for our Model's resource. Let's do it!

File: `app/code/local/Magentotutorial/Complexworld/Model/Resource/Eavblogpost.php`:

```
class Magentotutorial_Complexworld_Model_Resource_Eavblogpost extends Mage_Eav_Model_Entity_Abstract {
 protected function _construct()
 {
 $resource = Mage::getSingleton('core/resource');
 $this->setType('complexworld_eavblogpost');
 $this->setConnection(
 $resource->getConnection('complexworld_read'),
 $resource->getConnection('complexworld_write')
);
 }
}
```

So, already we're seeing a few differences between a simple Model Resource and an EAV Model Resource. First off, we're extending the `Mage_Eav_Model_Entity_Abstract` class. While `Mage_Eav_Model_Entity_Abstract` uses the same `_construct` concept as a regular Model Resource, there's no `_init` method. Instead, we need to handle the init ourselves. This means telling the resource what connection-resources it should use, and passing a unique identifier into the `setType` method of our object.

Another difference in `Mage_Eav_Model_Entity_Abstract` is `_construct` is **not** an abstract method, primarily for reasons of backwards compatibility with older versions of the system.

So, with that, let's clear the Magento cache and reload the page. You should see a new exception which reads

```
Invalid entity_type specified: complexworld_eavblogpost
```

Magento is complaining that it can't find a `entity_type` named `complexworld_eavblogpost`. This is the value you set above

```
$this->setType('complexworld_eavblogpost');
```

Every entity has a type. Types will, among other things, let the EAV system know which attributes a Model uses, and allow the system to link to tables that store the values for attributes. We'll need to let Magento know that we're adding a new entity type. Take a look in the MySQL table named `eav_entity_type`.

```
mysql> select * from eav_entity_type;

1. row ****
entity_type_id: 1
entity_type_code: customer
entity_model: customer/customer
attribute_model:
entity_table: customer/entity
value_table_prefix:
entity_id_field:
is_data_sharing: 1
data_sharing_key: default
default_attribute_set_id: 1
increment_model: eav/entity_increment_numeric
increment_per_store: 0
increment_pad_length: 8
increment_pad_char: 0

2. row ****
entity_type_id: 2
entity_type_code: customer_address
entity_model: customer/customer_address
attribute_model:
entity_table: customer/address_entity
value_table_prefix:
entity_id_field:
is_data_sharing: 1
data_sharing_key: default
default_attribute_set_id: 2
increment_model:
increment_per_store: 0
increment_pad_length: 8
increment_pad_char: 0
```

This table contains a list of all the `entity_types` in the system. The unique identifier `complexworld_eavblogpost` corresponds to the `entity_type_code` column.

## Systems and Applications

This illustrates the single most important Magento concept, one that many people struggle to learn.

Consider the computer in front of you. The OS (Mac OS X, Windows, Linux, etc.) is the software system. Your web browser (Firefox, Safari, IE, Opera) is the application. Magento **is a system** first, and an application second. You build eCommerce applications using the Magento system. What gets confusing is, there's a lot of places in Magento where the system code is exposed in a really raw form to the application code. The EAV system configuration living in the same database as your store's data is an example of this.

If you're going to get deep into Magento, you need to treat it like it's an old [Type 650](#) machine. That is to say, it's the kind of thing you can't effectively program applications in unless you have a deep understanding of the system itself.

## Creating a Setup Resource

So, it's theoretically possible to manually insert the rows you'll need into the Magento database to get your Model working, but it's not recommended. Fortunately, Magento provides a specialized [Setup Resource](#) that provides a number of helper method that will automatically create the needed records to get the system up and running.

So, for starters, configure the Setup Resource like you would any other.

```
<global>
<!-- ... -->
<resources>
 <complexworld_setup>
 <setup>
 <module>Magentotutorial_Complexworld</module>
 <class>Magentotutorial_Complexworld_Model_Resource_Setup</class>
 </setup>
 </complexworld_setup>
</resources>
<!-- ... -->
</global>
```

Next, create its class file.

File: app/code/local/Magentotutorial/Complexworld/Model/Resource/Setup.php:

```
class Magentotutorial_Complexworld_Model_Resource_Setup extends Mage_Eav_Mode
l_Entity_Setup {
```

Take note that we're extending from `Mage_Eav_Model_Entity_Setup` rather than `Mage_Core_Model_Resource_Setup`.

Finally, we'll set up our installer script. If you're not familiar with the naming conventions here, you'll want to review the [setup resource tutorial](#) on Setup Resources.

File: app/code/local/Magentotutorial/Complexworld/sql/complexworld\_setup/install-0.1.0.php:

```
<?php
$installer = $this;
throw new Exception("This is an exception to stop the installer from completing");
```

Clear your Magento Cache, reload your page, and the above exception should be thrown, meaning you've correctly configured your Setup Resource.

**NOTE:** We'll be building up our install script piece by piece. If you've read the [previous tutorial](#), you'll know you need to remove the setup's row from the `core_resource` table and clear your cache to make an installer script re-run. For the remainder of this tutorial, please remember that anytime we add or remove an item from our installer and re-run it, you'll need to remove this row from the database and clear your Magento cache. Normally you would create this file and run it once, a tutorial is something of an edge case.

## Adding the Entity Type

To begin, add the following to your Setup Resource installer script, and then run the script by loading any page (after removing the above exception)

```
$installer = $this;
$installer->startSetup();
$installer->addEntityType('complexworld_eavblogpost', array(
 //entity_mode is the URI you'd pass into a Mage::getModel() call
 'entity_model' => 'complexworld/eavblogpost',

 //table refers to the resource URI complexworld/eavblogpost
 //<complexworld_resource>...<eavblogpost><table>eavblog_posts</table>
 'table' =>'complexworld/eavblogpost',
));
$installer->endSetup();
```

We're calling the `addEntityType` method on our installer object. This method allows us to pass in the entity type (`complexworld_eavblogpost`) along with a list of parameters to set its default values. If you've run this script, you'll notice new rows in the `eav_attribute_group`, `eav_attribute_set`, and `eav_entity_type` tables.

So, with that in place, if we reload our `complexworld` page, we'll get a new error.

```
SQLSTATE[42S02]: Base table or view not found: 1146 Table 'magento.eavblog_po
sts' doesn't exist
```

## Creating the Data Tables

So, we've told Magento about our new entity type. Next, we need to add the MySQL tables that will be used to store all the entity values, as well as configure the system so it knows about these tables.

Our EAV Setup Resource has a method named `createEntityTables` which will automatically setup the tables we need, as well as add some configuration rows to the system. Let's add the following line to our setup resource.

```
$installer->createEntityTables(
 $this->getTable('complexworld/eavblogpost')
);
```

The `createEntityTables` method accepts two parameters. The first is the base table name, the second is a list of options. We're using the Setup Resource's `getTable` method to pull the table name from our config. If you've been following along, you know this should resolve to the string

`eavblog_posts`. We've omitted the second parameter which is an array of options you'll only need to used it for advanced situations that are beyond the scope of this tutorial.

After running the above script, you should have the following new tables in your database

```
eavblog_posts
eavblog_posts_char
eavblog_posts_datetime
eavblog_posts_decimal
eavblog_posts_int
eavblog_posts_text
eavblog_posts_varchar
```

You'll also have an additional row in the `eav_attribute_set` table

```
mysql> select * from eav_attribute_set order by attribute_set_id DESC LIMIT 1
\G
***** 1. row *****
attribute_set_id: 65
entity_type_id: 37
attribute_set_name: Default
sort_order: 6
```

So, let's go back to our page and reload.

```
http://example.com/complexworld
```

Success! You should see no errors or warnings, and a dumped `Magentotutorial_Complexworld_Model_Eavblogpost` --- with no data.

## Adding Attributes

The last step we need to take in our Setup Resource is telling Magento what attributes we want our EAV Model to have. This would be equivalent to adding new columns in a single database table setup. Again, the Setup Resource will help us. The method we're interested in is `addAttribute`.

The code from the previous section was simply telling Magento about a **type** of entity that we add to the system. These next bits of code are what will actually add possible attributes for our new type to the system.

We do that with the method `addAttribute`. When we call `addAttribute`, Magento will need to do several things to install your entities.

To start with, we'll give our `Eavblogpost` a single attribute named `title`.

```
/* ... */
$this->addAttribute('complexworld_eavblogpost', 'title', array(
 //the EAV attribute type, NOT a MySQL varchar
 'type' => 'varchar',
 'label' => 'Title',
 'input' => 'text',
 'class' => '',
 'backend' => '',
 'frontend' => '',
 'source' => '',
 'required' => true,
 'user_defined' => true,
 'default' => '',
 'unique' => false,
));
/* ... */
```

All right, that's a small pile of code. Let's break it apart.

The first argument to `addAttribute` is the entity type code. It has to match the code specified when calling `addEntityType`. It tells Magento which entity we are adding the attribute to, in our example it is our `complexworld_eavblogpost` entity. To see other available entities that come shipped with Magento, remember you can look into the `eav_entity_type` table at the `entity_type_code` column.

The second argument to `addAttribute` is the attribute code. It has to be unique within the given entity.

The third argument is where it get real interesting. This is an array of key value pairs, describing the attribute properties. For the sake of simplicity we've chose to define a single attribute, but you could go on to define as many as you'd like, by adding additional `addAttribute` calls to the setup script.

## Array of Key Value Pairs that Define the Attribute

Finally, we have a long list of attribute properties.

```
//the EAV attribute type, NOT a MySQL varchar
'type' => 'varchar',
'label' => 'Title',
'input' => 'text',
'class' => '',
'backend' => '',
'frontend' => '',
'source' => '',
'required' => true,
'user_defined' => true,
'default' => '',
'unique' => false,
```

Most of these define how Magento would build a backend form element for this attribute, and probably you'll won't have to deal with the.. That said, the one important property you'll want to make note of is

```
'type' => 'varchar'
```

This defines the type of the value that the attribute will contain. You'll recall that we added table for each attribute type

```
eavblog_posts_datetime
eavblog_posts_decimal
eavblog_posts_int
eavblog_posts_text
eavblog_posts_varchar
```

While these do not refer to the MySQL column types, (but instead the EAV attribute types), their names (varchar, datetime, etc.) are indicative of the values they'll hold.

All of these attribute properties are optional, if we wouldn't have specified them, Magento would have used a default value. These default values are defined in the `_prepareValues` method of the `Mage_Eav_Model_Entity_Setup` class (inherited by our setup class).

```
// Mage_Eav_Model_Entity_Setup
protected function _prepareValues($attr)
{
 $data = array(
 'backend_model' => $this->_getValue($attr, 'backend'),
 'backend_type' => $this->_getValue($attr, 'type', 'varchar'),
 'backend_table' => $this->_getValue($attr, 'table'),
 'frontend_model' => $this->_getValue($attr, 'frontend'),
 'frontend_input' => $this->_getValue($attr, 'input', 'text'),
 'frontend_label' => $this->_getValue($attr, 'label'),
 'frontend_class' => $this->_getValue($attr, 'frontend_class'),
 'source_model' => $this->_getValue($attr, 'source'),
 'is_required' => $this->_getValue($attr, 'required', 1),
 'is_user_defined'=> $this->_getValue($attr, 'user_defined', 0),
 'default_value' => $this->_getValue($attr, 'default'),
 'is_unique' => $this->_getValue($attr, 'unique', 0),
 'note' => $this->_getValue($attr, 'note'),
 'is_global' => $this->_getValue($attr, 'global',
 Mage_Catalog_Model_Resource_Eav_Attribute::S
COPE_GLOBAL
),
);
return $data;
}
```

The second argument to the method calls to `_getValue` is the array key from our `addAttribute` argument array, and the third is the default value. So by default Magento would assume you are adding a varchar attribute with a text input.

## Adding the other attributes

Lets add attributes for the blog post content and the post date. This is what the complete install script looks like.

```
$installer = $this;
```

```

$installer->startSetup();

$installer->addEntityType('complexworld_eavblogpost', array(
 //entity_mode is the URI you'd pass into a Mage::getModel() call
 'entity_model' => 'complexworld/eavblogpost',
 //table refers to the resource URI complexworld/eavblogpost
 //<complexworld_resource>...<eavblogpost><table>eavblog_posts</table>
 'table' =>'complexworld/eavblogpost',
));
;

$installer->createEntityTables(
 $this->getTable('complexworld/eavblogpost')
);

$this->addAttribute('complexworld_eavblogpost', 'title', array(
 //the EAV attribute type, NOT a MySQL varchar
 'type' => 'varchar',
 'label' => 'Title',
 'input' => 'text',
 'class' => '',
 'backend' => '',
 'frontend' => '',
 'source' => '',
 'required' => true,
 'user_defined' => true,
 'default' => '',
 'unique' => false,
));
;

$this->addAttribute('complexworld_eavblogpost', 'content', array(
 'type' => 'text',
 'label' => 'Content',
 'input' => 'textarea',
));
;

$this->addAttribute('complexworld_eavblogpost', 'date', array(
 'type' => 'datetime',
 'label' => 'Post Date',
 'input' => 'datetime',
 'required' => false,
));
;

$installer->endSetup();

```

So, now that we have everything in place, lets refresh things one last time to run our installer script. After calling `addAttribute`, we should have

1. A new row in `eav_entity_type` for the **complexworld\_eavblogpost** entity type
2. A new row in `eav_attribute` for the **title** attribute
3. A new row in `eav_attribute` for the **content** attribute
4. A new row in `eav_attribute` for the **date** attribute
5. A new row in `eav_entity_attribute`

## Tying it all Together

This is clearly the lamest.blogmodel.ever, but lets try adding some rows and iterating through a collection and get the heck out of here before our heads explode. Add the following two actions to your Index Controller.

```

public function populateEntriesAction() {
 for ($i=0;$i<10;$i++) {
 $weblog2 = Mage::getModel('complexworld/eavblogpost');
 $weblog2->setTitle('This is a test '.$i);
 $weblog2->setContent('This is test content '.$i);
 $weblog2->setDate(now());
 $weblog2->save();
 }

 echo 'Done';
}

public function showCollectionAction() {
 $weblog2 = Mage::getModel('complexworld/eavblogpost');
 $entries = $weblog2->getCollection()
 ->addAttributeToSelect('title')
 ->addAttributeToSelect('content');
}

```

```

$entries->load();
foreach($entries as $entry)
{
 // var_dump($entry->getData());
 echo '<h2>' . $entry->getTitle() . '</h2>';
 echo '<p>Date: ' . $entry->getDate() . '</p>';
 echo '<p>' . $entry->getContent() . '</p>';
}
echo '
Done
';
}

```

Let's populate some entries! Load up the following URL

```
http://magento.dev/index.php/complexworld/index/populateEntries
```

If you take a look at your database, you should see 10 new rows in the `eavblog_posts` table.

```

mysql> SELECT * FROM eavblog_posts ORDER BY entity_id DESC;
+-----+-----+-----+-----+-----+
| entity_id | entity_type_id | attribute_set_id | increment_id | parent_id |
| store_id | created_at | updated_at | is_active |
+-----+-----+-----+-----+
| 10 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
| 9 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
| 8 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
| 7 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
| 6 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
| 5 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
| 4 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
| 3 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
| 2 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
| 1 | 0 | 31 | 0 | 0 |
| 2009-12-06 08:36:41 | 2009-12-06 08:36:41 | 1 |
+-----+-----+-----+-----+

```

as well as 10 new rows in the `eavblog_posts_varchar` table.

```

mysql> SELECT * FROM eavblog_posts_varchar ORDER BY value_id DESC;
+-----+-----+-----+-----+-----+
| value_id | entity_type_id | attribute_id | store_id | entity_id | value |
+-----+-----+-----+-----+-----+
| 10 | 0 | 31 | 933 | 0 | 10 | This is a
| test 9 | 0 | 31 | 933 | 0 | 9 | This is a
| 9 | 0 | 31 | 933 | 0 | 9 | This is a
| test 8 | 0 | 31 | 933 | 0 | 8 | This is a
| 8 | 0 | 31 | 933 | 0 | 8 | This is a
| test 7 | 0 | 31 | 933 | 0 | 7 | This is a
| 7 | 0 | 31 | 933 | 0 | 7 | This is a
| test 6 | 0 | 31 | 933 | 0 | 6 | This is a
| 6 | 0 | 31 | 933 | 0 | 6 | This is a
| test 5 | 0 | 31 | 933 | 0 | 5 | This is a
| 5 | 0 | 31 | 933 | 0 | 5 | This is a
| test 4 | 0 | 31 | 933 | 0 | 4 | This is a
| 4 | 0 | 31 | 933 | 0 | 4 | This is a
| test 3 | 0 | 31 | 933 | 0 | 3 | This is a
| 3 | 0 | 31 | 933 | 0 | 3 | This is a
| test 2 | 0 | 31 | 933 | 0 | 2 | This is a
| 2 | 0 | 31 | 933 | 0 | 2 | This is a
| test 1 | 0 | 31 | 933 | 0 | 1 | This is a
| 1 | 0 | 31 | 933 | 0 | 1 | This is a
| test 0 | 0 | 31 | 933 | 0 | 0 | This is a
+-----+-----+-----+-----+-----+

```

Notice that eavblog\_posts\_varchar is linked to eavblog\_posts by the entity\_id column.

Finally, let's pull our Models back out. Load the following URL in your browser

```
http://magento.dev/index.php/complexworld/index/showCollection
```

This should give us a

```
Warning: include(Magentotutorial/Complexworld/Model/Resource/Eavblogpost/Collection.php) [function.include]:
failed to open stream: No such file or directory in /Users/username/Sites/
magento.dev/lib/Varien/Autoload.php on line 93
```

**So Close!** We didn't make a class for our collection object! Fortunately, doing so is just as easy as with a regular Model Resource. Add the following file with the following contents

File: Magentotutorial/Complexworld/Model/Resource/Eavblogpost/Collection.php:

```
class Magentotutorial_Complexworld_Model_Resource_Eavblogpost_Collection extends Mage_Eav_Model_Entity_Collection_Abstract
{
 protected function _construct()
 {
 $this->_init('complexworld/eavblogpost');
 }
}
```

This is just a standard Magento `_construct` method to initialize the Model. With this in place, reload the page, and we'll see all the titles and the content outputted. But notice, the date value is missing!

## Which Attributes?

Those of you with sharp eyes may have noticed something slightly different about the collection loading.

```
$entries = $weblog2->getCollection()
->addAttributeToSelect('title')
->addAttributeToSelect('content');
```

Because querying for EAV data can be SQL intensive, you'll need to specify which attributes it is you want your Models to fetch for you. This way the system can make only the queries it needs. If you're willing to suffer the performance consequences, you can use a wild card to grab **all** the attributes

```
$entries = $weblog2->getCollection()->addAttributeToSelect('*');
```

## Jumping Off

So, that should give you enough information to be dangerous, or at least enough information so you're not drowning the next time you're trying to figure out why the yellow shirts aren't showing up in your store. There's still plenty to learn about EAV; here's a few topics I would have liked to cover in greater detail, and may talk about in future articles

1. EAV Attributes: Attributes aren't limited to datetime, decimal, int, text and varchar. You can create your own class files to model different attributes. This is what the `attribute_model` entity property is for.
2. Collection Filtering: Filtering on EAV collections can get tricky, especially when you're dealing with the above mentioned non-simple attributes. You need to use the `addAttributeToFilter` method on your collection before loading.
3. The Magento EAV Hierarchy: Magento has taken their basic EAV Model and built up a hierarchy that's very tied to store functionality, as well as including strategies to reduce the number of queries an EAV Model generates (the concept of a **flat** Model, for example)

EAV Models are, without a doubt, the most complicated part of the Magento system that an ecommerce web developer will need to deal with. Remember to take deep breaths and that, at the end of the day, it's just programming. Everything happens for a concrete reason, you just need to figure out why.

22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Magento for Developers: Part 8—Varien Data Collections

by Alan Storm, updated for Magento 1.12

[Edit this page on GitHub](#) 

### Other articles in this series:

- [Part 1—Introduction to Magento](#)
- [Part 2—The Magento Config](#)
- [Part 3—Magento Controller Dispatch](#)
- [Part 4—Magento Layouts, Blocks and Templates](#)
- [Part 5—Magento Models and ORM Basics](#)
- [Part 6—Magento Setup Resources](#)
- [Part 7—Advanced ORM: Entity Attribute Value](#)
- [Part 8—Varien Data Collections](#)

Originally, as a PHP programmer, if you wanted to collect together a group of related variables you had one choice, the venerable `Array`. While it shares a name with C's array of memory addresses, a PHP array is a general purpose dictionary like object combined with the behaviors of a numerically indexed mutable array.

In other languages the choice isn't so simple. You have `multiple data structures` to chose from, each offering particular advantages in storage, speed and semantics. The PHP philosophy was to remove this choice from the client programmer and give them one useful data structure that was "good enough".

All of this is galling to a certain type of software developer, and PHP 5 set out to change the status quo by offering built-in classes and interfaces that allow you to create your own data structures.

```
$array = new ArrayObject();
class MyCollection extends ArrayObject{...}
$collection = new MyCollection();
$collection[] = 'bar';
```

While this is still galling to a certain type of software developer, as you don't have access to low level implementation details, you do have the ability to create array-like Objects with methods that encapsulate specific functionality. You can also setup rules to offer a level of type safety by only allowing certain kinds of Objects into your Collection.

It should come as no surprise that Magento offers you a number of these Collections. In fact, every Model object that follows the Magento interfaces gets a Collection type for free. Understanding how these Collections work is a key part to being an effective Magento programmer. We're going to take a look at Magento Collections, starting from the bottom and working our way up. Set up a `controller action` where you can run arbitrary code, and let's get started.

# A Collection of Things

First, we're going to create a few new Objects.

```
$thing_1 = new Varien_Object();
$thing_1->setName('Richard');
$thing_1->setAge(24);

$thing_2 = new Varien_Object();
$thing_2->setName('Jane');
$thing_2->setAge(12);

$thing_3 = new Varien_Object();
$thing_3->setName('Spot');
$thing_3->setLastName('The Dog');
$thing_3->setAge(7);
```

The `Varien_Object` class defines the object all Magento Models inherit from. This is a common pattern in object oriented systems, and ensures you'll always have a way to easily add methods/functionality to **every** object in your system without having to edit every class file.

Any Object that extends from `Varien_Object` has magic getter and setters that can be used to set data properties. Give this a try

```
var_dump($thing_1->getName());
```

If you don't know what the property name you're after is, you can pull out all the data as an array

```
var_dump($thing_3->getData());
```

The above will give you an array something like

```
array
'name' => string 'Spot' (length=4)
'last_name' => string 'The Dog' (length=7)
'age' => int 7
```

Notice the property named "last\_name"? If there's an underscore separated property, you camel case it if you want to use the getter and setter magic.

```
$thing_1->setLastName('Smith');
```

The ability to do these kinds of things is part of the power of PHP5, and the development style a certain class of people mean when they say "Object Oriented Programming".

So, now that we have some Objects, let's add them to a Collection. Remember, a Collection is like an Array, but is defined by a PHP programmer.

```
$collection_of_things = new Varien_Data_Collection();
$collections
 ->addItem($thing_1)
 ->addItem($thing_2)
 ->addItem($thing_3);
```

The `Varien_Data_Collection` is the Collection that most Magento data Collections inherit from. Any method you can call on a `Varien_Data_Collection` you can call on Collections higher up the chain (We'll see more of this later)

What can we do with a Collection? For one, we can use `foreach` to iterate over it

```
foreach($collection_of_things as $thing)
{
 var_dump($thing->getData());
}
```

There are also shortcuts for pulling out the first and last items

```
var_dump($collection_of_things->getFirstItem()->getData());
var_dump($collection_of_things->getLastItem()->getData());
```

Want your Collection data as XML? There's a method for that

```
var_dump($collection_of_things->toXml());
```

Only want a particular field?

```
var_dump($collection_of_things->getColumnValues('name'));
```

The team at Magento have even given us some rudimentary filtering capabilities.

```
var_dump($collection_of_things->getItemsByColumnValue('name', 'Spot'));
```

Neat stuff.

## Model Collections

So, this is an interesting exercise, but why do we care?

We care because all of Magento's built in data Collections inherit from this object. That means if you have, say, a product Collection you can do the same sort of things. Let's take a look

```
public function testAction()
{
 $collection_of_products = Mage::getModel('catalog/product')->getCollection();
 var_dump($collection_of_products->getFirstItem()->getData());
}
```

Most Magento Model objects have a method named `getCollection` which will return a collection that, by default, is initialized to return every Object of that type in the system.

**A Quick Note:** Magento's Data Collections contain a lot of complicated logic that handles when to use an index or cache, as well as the logic for the EAV entity system. Successive method calls to the same Collection over its life can often result in unexpected behavior. Because of that, all the of the following examples are wrapped in a single method action. I'd recommend doing the same while you're experimenting. Also, [XDebug's](#) `var_dump` is a godsend when working with Magento Objects and Collections, as it will (usually) intelligently short circuit showing hugely recursive Objects, but still display a useful representation of the Object structure to you.

The products Collection, as well as many other Magento Collections, also have the `Varien_Data_Collection_Db` class in their ancestor chain. This gives us a lot of useful methods. For example, if you want to see the select statement your Collection is using

```
public function testAction()
{
 $collection_of_products = Mage::getModel('catalog/product')->getCollection();
 var_dump($collection_of_products->getSelect()); //might cause a segmentation fault
}
```

The output of the above will be

```
object(Varien_Db_Select)[94]
protected '_bind' =>
array
empty
protected '_adapter' =>
...
...
```

Whoops! Since Magento is using the Zend database abstraction layer, your Select is also an Object. Let's see that as a more useful string.

```
public function testAction()
{
 $collection_of_products = Mage::getModel('catalog/product')->getCollection();
 //var_dump($collection_of_products->getSelect()); //might cause a segmentation fault
 var_dump(
 (string) $collection_of_products->getSelect()
);
}
```

Sometimes this is going to result in a simple select

```
'SELECT `e`.* FROM `catalog_product_entity` AS `e`'
```

Other times, something a bit more complex

```
string 'SELECT `e`.*, `price_index`.`price`, `price_index`.`final_price`, IF(`price_index`.`tier_price`,
LEAST(`price_index`.`min_price`, `price_index`.`tier_price`), `price_index`.`min_price`) AS `minimal_price`,
'price_index`.`min_price`, `price_index`.`max_price`, `price_index`.`tier_price` FROM `catalog_product_entity`
AS `e` INNER JOIN `catalog_product_index_price` AS `price_index` ON price_index.entity_id = e.entity_id AND
price_index.website_id = '1' AND price_index.customer_group_id = 0'
```

The discrepancy depends on which attributes you're selecting, as well as the aforementioned indexing and cache. If you've been following along with the other articles in this series, you know that many Magento models (including the Product Model) use an [EAV](#) system. By default, a EAV Collection will not include all of an Object's attributes. You can add them all by using the `addAttributeToSelect` method

```
$collection_of_products = Mage::getModel('catalog/product')
 ->getCollection()
 ->addAttributeToSelect('*'); //the asterisk is like a SQL SELECT * FROM
...
```

Or, you can add just one

```
//or just one
$collection_of_products = Mage::getModel('catalog/product')
 ->getCollection()
 ->addAttributeToSelect('meta_title');
```

or chain together several

```
//or just one
$collection_of_products = Mage::getModel('catalog/product')
 ->getCollection()
 ->addAttributeToSelect('meta_title')
 ->addAttributeToSelect('price');
```

## Lazy Loading

One thing that will trip up PHP developers new to Magento's ORM system is **when** Magento makes its database calls. When you're writing literal SQL, or even when you're using a basic ORM system, SQL calls are often made immediately when instantiating an Object.

```
$model = new Customer();
//SQL Calls being made to Populate the Object
echo 'Done'; //execution continues
```

Magento doesn't work that way. Instead, the concept of [Lazy Loading](#) is used. In simplified terms, Lazy loading means that no SQL calls are made until the client-programmer needs to access the data. That means when you do something like this

```
$collection_of_products = Mage::getModel('catalog/product')
 ->getCollection();
```

Magento actually hasn't gone out to the database yet. You can safely add attributes later

```
$collection_of_products = Mage::getModel('catalog/product')
 ->getCollection();
$collection_of_products->addAttributeToSelect('meta_title');
```

and not have to worry that Magento is making a database query each time a new attribute is added. The database query will not be made until you attempt to access an item in the Collection.

In general, try not to worry too much about the implementation details in your day to day work. It's good to know that there's a SQL backend and Magento is doing SQLy things, but when you're coding up a feature try to forget about it, and just treat the objects as black boxes that do what you need.

## Filtering Database Collections

The most important method on a database Collection is `addFieldToFilter`. This adds your `WHERE` clauses to the SQL query being used behind the scenes. Consider this bit of code, run against the sample data database (substitute your own SKU if you're using a different set of product data)

```
public function testAction()
{
 $collection_of_products = Mage::getModel('catalog/product')
 ->getCollection();
 $collection_of_products->addFieldToFilter('sku', 'n2610');

 //another neat thing about collections is you can pass them into the count
 //function. More PHP5 powered goodness
 echo "Our collection now has " . count($collection_of_products) . ' items';
 var_dump($collection_of_products->getFirstItem()->getData());
}
```

The first parameter of `addFieldToFilter` is the attribute you wish to filter by. The second is the value you're looking for. Here's we're adding a `sku` filter for the value `n2610`.

The second parameter can also be used to specify the **type** of filtering you want to do. This is where things get a little complicated, and worth going into with a little more depth.

So by default, the following

```
$collection_of_products->addFieldToFilter('sku', 'n2610');
```

is (essentially) equivalent to

```
WHERE sku = "n2610"
```

Take a look for yourself. Running the following

```
public function testAction()
{
 var_dump(
 (string)
 Mage::getModel('catalog/product')
 ->getCollection()
 ->addFieldToFilter('sku', 'n2610')
 ->getSelect());
}
```

will yield

```
SELECT `e`.* FROM `catalog_product_entity` AS `e` WHERE (e.sku = 'n2610')
```

Keep in mind, this can get complicated fast if you're using an EAV attribute. Add an attribute

```
var_dump(
 (string)
 Mage::getModel('catalog/product')
 ->getCollection()
 ->addAttributeToSelect('*')
 ->addFieldToFilter('meta_title', 'my title')
 ->getSelect()
);
```

and the query gets gnarly.

```
SELECT `e`.*, IF(_table_meta_title.value_id>0, _table_meta_title.value, _table_meta_title_default.value) AS `meta_title`
FROM `catalog_product_entity` AS `e`
INNER JOIN `catalog_product_entity_varchar` AS `_table_meta_title_default`
 ON (_table_meta_title_default.entity_id = e.entity_id) AND (_table_meta_title_default.attribute_id='103')
 AND _table_meta_title_default.store_id=0
LEFT JOIN `catalog_product_entity_varchar` AS `_table_meta_title`
 ON (_table_meta_title.entity_id = e.entity_id) AND (_table_meta_title.attribute_id='103')
 AND (_table_meta_title.store_id='1')
WHERE (IF(_table_meta_title.value_id>0, _table_meta_title.value, _table_meta_title_default.value) = 'my title')
```

Not to belabor the point, but try not to think too much about the SQL if you're on deadline.

## Other Comparison Operators

I'm sure you're wondering "what if I want something other than an equals by query"? Not equal, greater than, less than, etc. The `addFieldToFilter` method's second parameter has you covered there as well. It supports an alternate syntax where, instead of passing in a string, you pass in a single element Array.

The key of this array is the **type** of comparison you want to make. The value associated with that key is the value you want to filter by. Let's redo the above filter, but with this explicit syntax

```
public function testAction()
{
 var_dump(
 (string)
 Mage::getModel('catalog/product')
 ->getCollection()
 ->addFieldToFilter('sku', array('eq'=>'n2610'))
 ->getSelect()
);
}
```

Calling out our filter

```
addFieldToFilter('sku',array('eq'=>'n2610'))
```

As you can see, the second parameter is a PHP Array. Its key is `eq`, which stands for *equals*. The value for this key is `n2610`, which is the value we're filtering on.

Magento has a number of these english language like filters that will bring a [tear of remembrance](#) (and perhaps pain) to any old perl developers in the audience.

Listed below are all the filters, along with an example of their SQL equivalents.

```
array("eq"=>'n2610')
WHERE (e.sku = 'n2610')

array("neq"=>'n2610')
WHERE (e.sku != 'n2610')

array("like"=>'n2610')
WHERE (e.sku like 'n2610')

array("nlike"=>'n2610')
WHERE (e.sku not like 'n2610')

array("is"=>'n2610')
WHERE (e.sku is 'n2610')

array("in"=>array('n2610'))
WHERE (e.sku in ('n2610'))

array("nin"=>array('n2610'))
WHERE (e.sku not in ('n2610'))

array("notnull"=>true)
WHERE (e.sku is NOT NULL)

array("null"=>true)
WHERE (e.sku is NULL)

array("gt"=>'n2610')
WHERE (e.sku > 'n2610')

array("lt"=>'n2610')
WHERE (e.sku < 'n2610')

array("gteq"=>'n2610')
WHERE (e.sku >= 'n2610')

array("moreq"=>'n2610') //a weird, second way to do greater than equal (Doesn't work on > 1.8 CE EDITION do the same as eq)
WHERE (e.sku >= 'n2610')

array("lteq"=>'n2610')
WHERE (e.sku <= 'n2610')
```

```

array("finset"=>array('n2610'))
WHERE (find_in_set('n2610', e.sku))

array('from'=>'10', 'to'=>'20')
WHERE e.sku >= '10' and e.sku <= '20'

```

Most of these are self explanatory, but a few deserve a special callout

## in, nin, find\_in\_set

The in, nin and finset conditionals allow you to pass in an Array of values. That is, the value portion of your filter array is itself allowed to be an array.

```

array("in"=>array('n2610', 'ABC123')
WHERE (e.sku in ('n2610', 'ABC123'))

```

## notnull, null

The keyword NULL is special in most flavors of SQL. It typically won't play nice with the standard equality (=) operator. Specifying notnull or null as your filter type will get you the correct syntax for a NULL comparison while ignoring whatever value you pass in

```

array("notnull"=>true)
WHERE (e.sku is NOT NULL)

```

## from - to filter

This is another special format that breaks the standard rule. Instead of a single element array, you specify a two element array. One element has the key from, the other element has the key to. As the keys indicated, this filter allows you to construct a from/to range without having to worry about greater than and less than symbols

```

public function testAction()
{
 var_dump(
 (string)
 Mage::getModel('catalog/product')
 ->getCollection()
 ->addFieldToFilter('price',array('from'=>'10', 'to'=>'20'))
 ->getSelect()
);
}

```

The above yields

```

WHERE (_table_price.value >= '10' and _table_price.value <= '20')

```

## AND or OR, or is that OR and AND?

Finally, we come to the boolean operators. It's the rare moment where we're only filtering by one attribute. Fortunately, Magento's Collections have us covered. You can chain together multiple calls to addFieldToFilter to get a number of "AND" queries.

```

function testAction()
{
 echo
 (string)
 Mage::getModel('catalog/product')
 ->getCollection()
 ->addFieldToFilter('sku',array('like'=>'a%'))
 ->addFieldToFilter('sku',array('like'=>'b%'))
 ->getSelect();
}

```

By chaining together multiple calls as above, we'll produce a where clause that looks something like the following

```

WHERE (e.sku like 'a%') AND (e.sku like 'b%')

```

To those of you that just raised your hand, yes, the above example would always return 0 records. No sku can begin with BOTH an a and a b. What we probably want here is an OR query. This brings us to another confusing aspect of addFieldToFilter's second parameter.

If you want to build an OR query, you need to pass an Array of filter Arrays in as the second parameter. I find it's best to assign your individual filter Arrays to variables

```
public function testAction()
{
 $filter_a = array('like'=>'a%');
 $filter_b = array('like'=>'b%');
}
```

and then assign an array of all my filter variables

```
public function testAction()
{
 $filter_a = array('like'=>'a%');
 $filter_b = array('like'=>'b%');
 echo
 (string)
 Mage::getModel('catalog/product')
 ->getCollection()
 ->addFieldToFilter('sku', array($filter_a, $filter_b))
 ->getSelect();
}
```

In the interest of being explicit, here's the aforementioned array of filter arrays.

```
array($filter_a, $filter_b)
```

This will give us a WHERE clause that looks something like the following

```
WHERE (((e.sku like 'a%') or (e.sku like 'b%')))
```

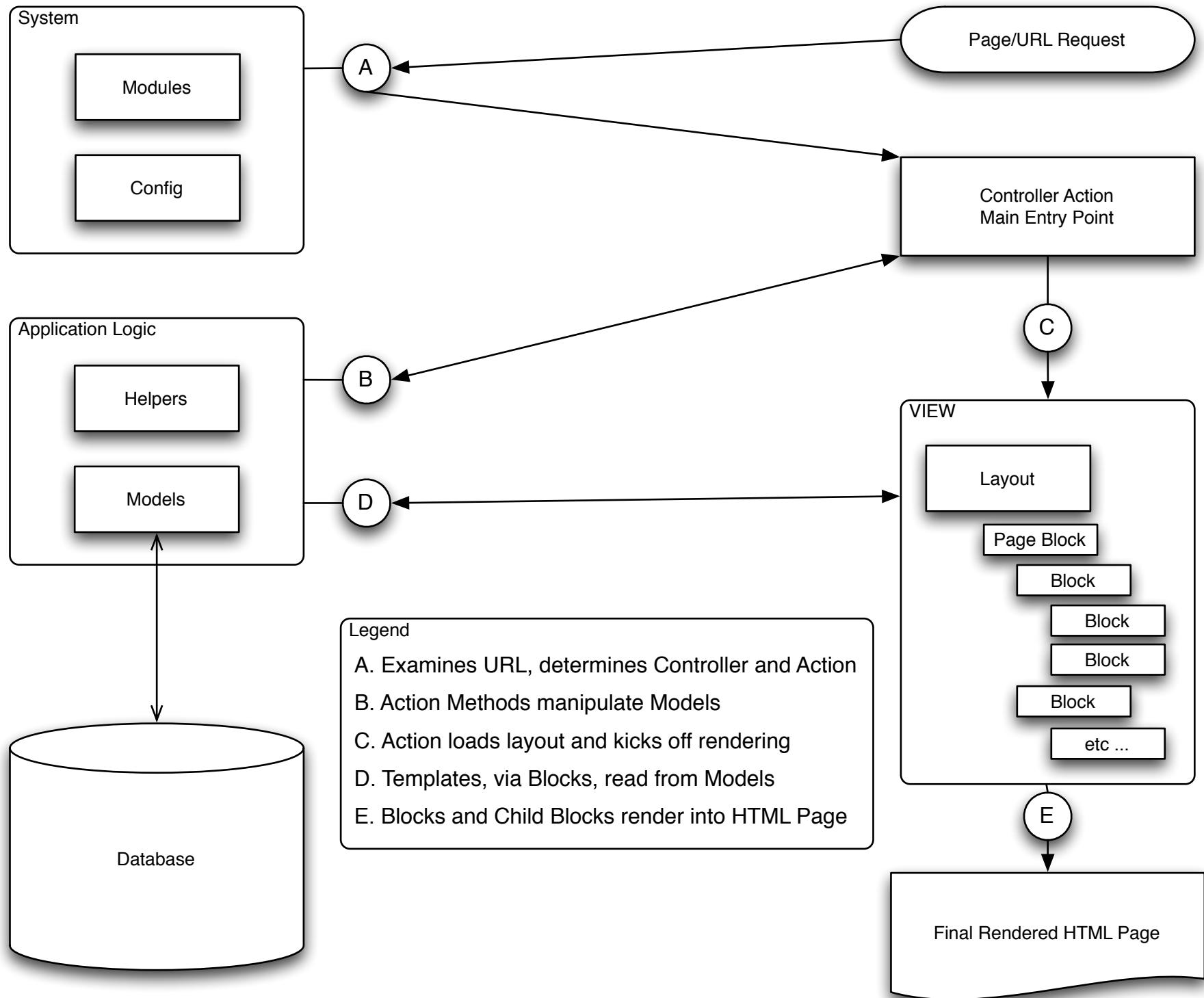
## Wrap Up

You're now a Magento developer walking around with some serious firepower. Without having to write a single line of SQL you now know how to query Magento for any Model your store or application might need.

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.



## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Introduction to the Magento 1.x REST API

[Edit this page on GitHub](#) 

## Introduction

### Authentication

### Common HTTP Status Codes

### HTTP Methods

### GET Filters

### Permission Settings

### Resources

### Response Formats

### Testing REST Resources

## Introduction to API

Overview

Overall Capabilities

Authentication

Output Formats

HTTP Verbs

Request Structure

REST Resources

Products

Product Categories

Product Images

Product Websites

Customers

Customer Addresses

Inventory

Sales Orders

Order Items

Order Addresses

Order Comments

Preparing for REST API

Preparing REST API for the Third-Party Application

PHP Examples

Create a simple product as an Admin user with OAuth authentication

Retrieve the list of products as a Customer user with OAuth authentication

REST Client Example

# Introduction to API

## Overview

What is REST API? To make it simple, REST API defines a set of functions to which the developers can perform requests and receive responses. The interaction is performed via the HTTP protocol. An advantage of such an approach is the wide usage of HTTP. That is why REST API can be used practically for any programming language.

Common characteristics of Magento REST API resources are as follows: (**magentohost** is your domain)

- You access the resource by sending an HTTP request to the Magento API server. The server replies with a response that contains either the data you requested, or the status indicator, or even both.
- All resources are located at <https://magentohost/api/rest/>.
- All resources may return different HTTP status codes (e.g., HTTP Status Code 200 for success response or HTTP Status Code 400 for the bad request).
- You request a particular resource by adding a particular path to the base URL that specifies the resource.

## Overall Capabilities

Magento REST API allows managing a number of features, namely:

- Managing customers.
- Managing customer addresses.
- Managing products.
- Retrieving sales orders.
- Managing inventory.

## Authentication

Magento REST API uses 3-legged [OAuth 1.0a](#) protocol to authenticate the application to access the Magento service.

## Output Formats

The REST API supports the response in two formats, which are XML and JSON.

## HTTP Verbs

HTTP verbs are used to manage the state of resources. In Magento REST API, there are four verbs used to manage resources: GET, POST, PUT, and DELETE. You can get the contents of the data using HTTP GET, delete the data using HTTP DELETE, and create or update the data using POST/PUT.

## Request Structure

All URLs in REST API have the following base URL.

<http://magentohost/api/rest/>

### Example

Supposing, you want to retrieve the list of customers from Magento. To do this, you need to use the GET HTTP method. The GET request to retrieve the list of customers will look as follows:

<http://magentohost/api/rest/customers>

where

- `http://magentohost/api/rest/` - endpoint
- `/customers` - action URL

## REST Resources

The Magento REST API allows you to manage customers, customer addresses, sales orders, inventory, and products. REST API is organized into the following categories:

### Products

- Retrieve the list of products, create, update, and delete a product.
- Resource Structure:** `http://magentohost/api/rest/products`

### Product Categories

- Retrieve the list of categories assigned to a product, assign, and unassign the category to/from the specific product.
- Resource Structure:** `http://magentohost/api/rest/products/:productId/categories`

### Product Images

- Retrieve the list of images assigned to a product, add, update, and remove an image to/from the specific product.
- Resource Structure:** `http://magentohost/api/rest/products/:productId/images`

### Product Websites

- Retrieve the list of websites assigned to a product, assign, and unassign a website to/from the specific product.
- Resource Structure:** `http://magentohost/api/rest/products/:productId/websites`

### Customers

- Retrieve the list of customers, create, delete a customer, and update the customer information.
- Resource Structure:** `http://magentohost/api/rest/customers`

### Customer Addresses

- Retrieve the list of customer addresses, create, update, and delete the customer address.
- Resource Structure:** `http://magentohost/api/rest/customers/:customerId/addresses`

### Inventory

- Retrieve the list of stock items and update required stock items.
- Resource Structure:** `http://magentohost/api/rest/stockitems`

### Sales Orders

- Retrieve the list of sales orders as well as the specific order information.
- Resource Structure:** `http://magentohost/api/rest/orders`

### Order Items

- Retrieve order items for the specific order.
- Resource Structure:** `http://magentohost/api/rest/orders/:orderId/items`

### Order Addresses

- Retrieve information on order billing and shipping addresses for the specific order.
- Resource Structure:** <http://magentohost/api/rest/orders/:orderid/addresses>

#### Order Comments

- Retrieve order comments for the specific order

**Resource Structure:** <http://magentohost/api/rest/orders/:orderid/comments>

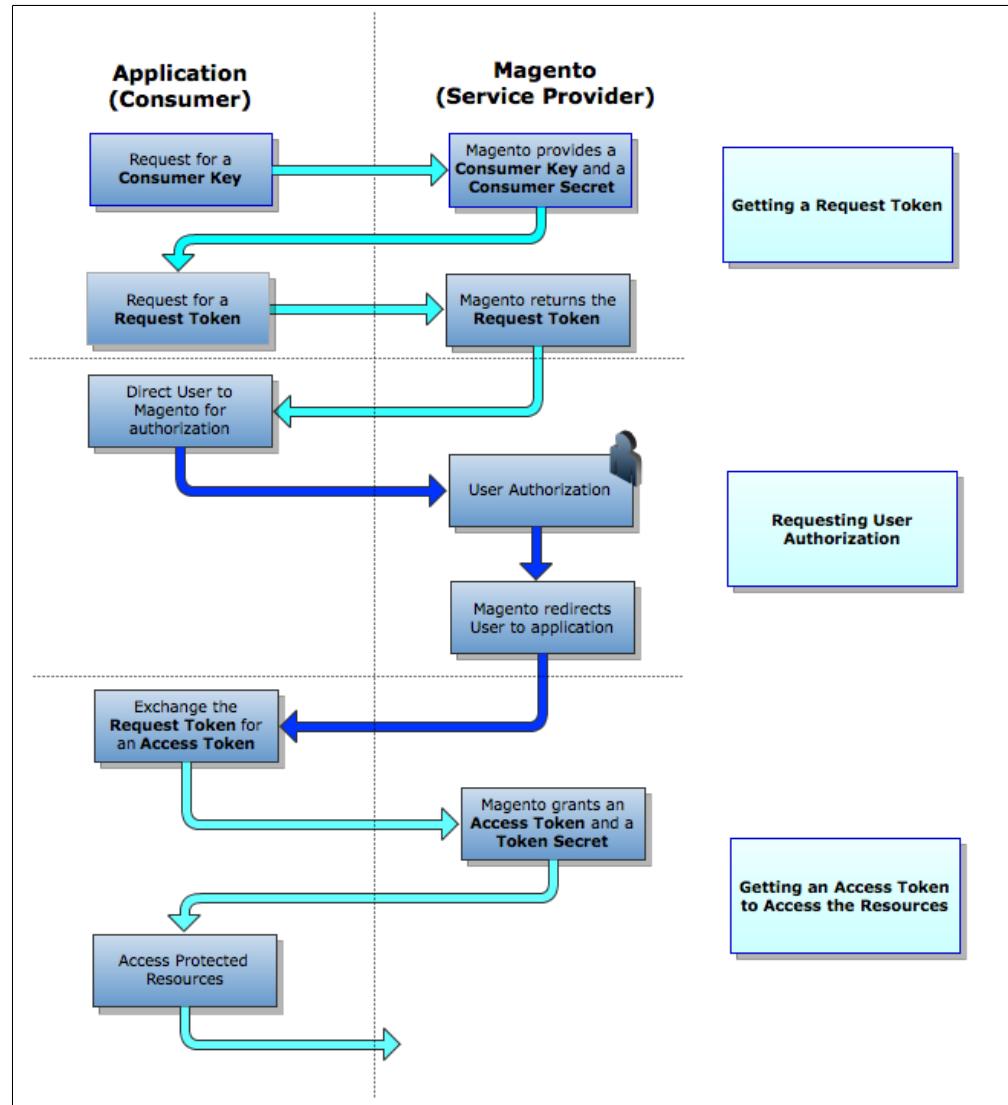
## Preparing for REST API

These steps are required for utilizing REST API resources:

1. Set up permissions for REST resource operations from Magento Admin Panel.
2. Configure the attributes for different users types in Magento Admin Panel. There are 3 different types of users in accessing the data: Admin, Customer, and Guest. Admin is the backend logged in user, Customer is the fronted logged in user, and Guest is a non-logged in fronted user.

## Preparing REST API for the Third-Party Application

1. Register the third-party application (Consumer) in Magento Admin Panel.
2. The third-party application will utilize the provided consumer credentials to call Magento store for getting the access token to access the data.



# PHP Examples

## Create a simple product as an Admin user with OAuth authentication

```
<?php
/**
 * Example of simple product POST using Admin account via Magento REST API. OAuth
 * authorization is used.
 *
 * This file is a stand-alone OAuth client PHP file, which handles everything with the
 * OAuth three-legged authentication.
 * It uses PHP session to persist the current authentication state (step), the OAuth
 * request token with its secret
 * and the OAuth access token with its secret.
 *
 * Create this file oauth_admin.php in your Magento 1.x instance root folder to run
 * this oauth authentication example.
 *
 * oauth_admin.php
 */
$callbackUrl = "http://yourhost/oauth_admin.php";
$temporaryCredentialsRequestUrl = "http://magentohost/oauth/initiate?oauth_callback=" .
urlencode($callbackUrl);
$adminAuthorizationUrl = 'http://magentohost/admin/oauth_authorize';
$accessTokenRequestUrl = 'http://magentohost/oauth/token';
$apiUrl = 'http://magentohost/api/rest';
$consumerKey = 'yourconsumerkey';
$consumerSecret = 'yourconsumersecret';

session_start();
if (!isset($_GET['oauth_token']) && !isset($_SESSION['state']) && $_SESSION['state'] ==
1) {
 $_SESSION['state'] = 0;
}
try {
 $authType = ($_SESSION['state'] == 2) ? OAUTH_AUTH_TYPEORIZATION :
OAUTH_AUTH_TYPE_URI;
 $oauthClient = new OAuth($consumerKey, $consumerSecret, OAUTH_SIG_METHOD_HMACSHA1,
$authType);
 $oauthClient->enableDebug();

 if (!isset($_GET['oauth_token']) && !$SESSION['state']) {
 // step 1 (state 1) - Get the initial temporary request token.
 $requestToken = $oauthClient-
>getRequestToken($temporaryCredentialsRequestUrl);

 // persist the request token secret in the session variable for step 3 to get
 the access token and secret.
 $_SESSION['secret'] = $requestToken['oauth_token_secret'];
 $_SESSION['state'] = 1;

 // step 2 (state 2) - redirect to the Oauth admin authorization url to
 validate and confirm / reject the admin Oauth authorization request.
 // variable $requestToken['oauth_token'] has the temporary request token
 header('Location: ' . $adminAuthorizationUrl . '?oauth_token=' .
$requestToken['oauth_token']);
 exit;
 } elseif ($_SESSION['state'] == 1) {
 //step 3 (state 3) Exchange the temporary request token and secret to get the
 final access token and secret.
 }
}
```

```

$oauthClient->setToken($_GET['oauth_token'], $_SESSION['secret']);
$accessToken = $oauthClient->getAccessToken($accessTokenRequestUrl);

// persist the access token and its secret in the session variable
$_SESSION['state'] = 2;
$_SESSION['token'] = $accessToken['oauth_token'];
$_SESSION['secret'] = $accessToken['oauth_token_secret'];

// redirect back to the callback url which is the same file
header('Location: ' . $callbackUrl);
exit;
} else {

 // send a POST request to create a simple product
 $oauthClient->setToken($_SESSION['token'], $_SESSION['secret']);
 $resourceUrl = "$apiUrl/products";
 $productData = json_encode(array(
 'type_id' => 'simple',
 'attribute_set_id' => 4,
 'sku' => 'simple' . uniqid(),
 'weight' => 1,
 'status' => 1,
 'visibility' => 4,
 'name' => 'Simple Product',
 'description' => 'Simple Description',
 'short_description' => 'Simple Short Description',
 'price' => 99.95,
 'tax_class_id' => 0,
)));
 $headers = array('Content-Type' => 'application/json');
 $oauthClient->fetch($resourceUrl, $productData, OAUTH_HTTP_METHOD_POST,
$headers);
 print_r($oauthClient->getLastResponseInfo());
}
} catch (OAuthException $e) {
 print_r($e);
}

```

## Retrieve the list of products as a Customer user with OAuth authentication

```

<?php
/**
 * Example of products list retrieve using Customer account via Magento REST API. OAuth
authorization is used
*
* This file is a stand-alone Oauth client PHP file, which handles everything with the
Oauth three-legged authentication.
* It uses PHP session to persist the current authentication state (step), the oauth
request token with its secret
* and the oauth access token with its secret.
*
* Create this file oauth_customer in your Magento 1.x instance root folder to run this
oauth authentication example.
*
* oauth_customer.php
*/
$callbackUrl = "http://yourhost/oauth_customer.php";
$temporaryCredentialsRequestUrl = "http://magentohost/oauth/initiate?oauth_callback=" .
urlencode($callbackUrl);

```

```

$adminAuthorizationUrl = 'http://magentohost/oauth/authorize';
$accessTokenRequestUrl = 'http://magentohost/oauth/token';
$apiUrl = 'http://magentohost/api/rest';
$consumerKey = 'yourconsumerkey';
$consumerSecret = 'yourconsumersecret';

session_start();
if (!isset($_GET['oauth_token']) && isset($_SESSION['state'])) && $_SESSION['state'] == 1) {
 $_SESSION['state'] = 0;
}
try {
 $authType = ($_SESSION['state'] == 2) ? OAUTH_AUTH_TYPE_AUTHORIZATION : OAUTH_AUTH_TYPE_URI;
 $oauthClient = new OAuth($consumerKey, $consumerSecret, OAUTH_SIG_METHOD_HMACSHA1, $authType);
 $oauthClient->enableDebug();

 if (!isset($_GET['oauth_token']) && !$SESSION['state']) {
 // step 1 (state 1) - Get the initial temporary request token.
 $requestToken = $oauthClient->getRequestToken($temporaryCredentialsRequestUrl);

 // persist the request token secret in the session variable for step 3 to get the access token and secret.
 $_SESSION['secret'] = $requestToken['oauth_token_secret'];
 $_SESSION['state'] = 1;

 // step 2 (state 2) - redirect to the Oauth customer authorization url.
 // variable $requestToken['oauth_token'] has the temporary request token
 header('Location: ' . $adminAuthorizationUrl . '?oauth_token=' . $requestToken['oauth_token']);
 exit;
 } elseif ($_SESSION['state'] == 1) {
 //step 3 (state 3) Exchange the temporary request token and secret to get the final access token and secret.
 $oauthClient->setToken($_GET['oauth_token'], $_SESSION['secret']);
 $accessToken = $oauthClient->getAccessToken($accessTokenRequestUrl);

 // persist the access token and its secret in the session variable to access the products resource
 $_SESSION['state'] = 2;
 $_SESSION['token'] = $accessToken['oauth_token'];
 $_SESSION['secret'] = $accessToken['oauth_token_secret'];

 // redirect back to the callback url which is the same file
 header('Location: ' . $callbackUrl);
 exit;
 } else {
 // send a GET request to list all the products
 $oauthClient->setToken($_SESSION['token'], $_SESSION['secret']);
 $resourceUrl = "$apiUrl/products";
 $oauthClient->fetch($resourceUrl, array(), 'GET', array('Content-Type' => 'application/json', 'Accept' => '*/*'));
 $productsList = json_decode($oauthClient->getLastResponse());
 print_r($productsList);
 }
} catch (OAuthException $e) {
 print_r($e);
}

```

## REST Client Example

## Retrieving the list of Products as a Guest

1. Use the [REST Client](#) that is a Firefox add-on. In the REST Client, in the **Method** drop-down list, select the **GET** option.
2. In the URL field, enter the following URL: <http://magentohost/api/rest/products?limit=2>.
3. Click **Send**. Information about the products will be displayed in the response body. Example in the XML format is as follows:

### Example: XML

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <entity_id>16</entity_id>
 <type_id>simple</type_id>
 <sku>n2610</sku>
 <description>The Nokia 2610 is an easy to use device that combines multiple messaging options including email, instant messaging, and more. You can even download MP3 ringtones, graphics, and games straight to the phone, or surf the Internet with Cingular's MEdia Net service. It's the perfect complement to Cingular service for those even remotely interested in mobile Web capabilities in an affordable handset.
 Design
 Compact and stylish, the 2610 features a candybar design sporting a bright 128 x 128 pixel display capable of displaying over 65,000 colors. Most of the phone's features and on-screen menus are controlled by a center toggle on the control pad. A standard hands-free headphone jack is provided, as are volume control keys, and there's even a "Go-To" button that can be assigned by the user for quick access to favorite applications. Lastly, the included speakerphone allows you to talk handsfree, and because the phone sports an internal antenna, there's nothing to snag or break off.
 </description>
 <meta_keyword>Nokia 2610, cell, phone, </meta_keyword>
 <short_description>The words "entry level" no longer mean "low-end," especially when it comes to the Nokia 2610. Offering advanced media and calling features without breaking the bank</short_description>
 <name>Nokia 2610 Phone</name>
 <meta_title>Nokia 2610</meta_title>
 <meta_description>Offering advanced media and calling features without breaking the bank, The Nokia 2610 is an easy to use</meta_description>
 <regular_price_with_tax>149.99</regular_price_with_tax>
 <regular_price_without_tax>149.99</regular_price_without_tax>
 <final_price_with_tax>149.99</final_price_with_tax>
 <final_price_without_tax>149.99</final_price_without_tax>
 <is_saleable>1</is_saleable>
 <image_url>http://magentohost/imageulr/nokia.jpg</image_url>
 </data_item>
 <data_item>
 <entity_id>17</entity_id>
 <type_id>simple</type_id>
 <sku>bb8100</sku>
 <description> Like the BlackBerry 7105t, the BlackBerry 8100 Pearl is
 The BlackBerry 8100 Pearl sports a large 240 x 260 screen that supports over 65,000 colors-- plenty of real estate to view your e-mails, Web browser content, messaging sessions, and attachments. The venerable BlackBerry trackwheel has been replaced on this model with an innovative four-way trackball placed below the screen. On the rear of the handheld, you'll find a 1.3-megapixel camera and a self portrait mirror. The handheld's microSD memory card slot is located inside the device, behind the battery. There's also a standard 2.5mm headset jack that can be used with the included headset, as well as a mini-USB port for data connectivity.</description>
 <meta_keyword>Blackberry, 8100, pearl, cell, phone</meta_keyword>
 <short_description>The BlackBerry 8100 Pearl is a departure from the form factor of previous BlackBerry devices. This BlackBerry handset is far more phone-like, and RIM's engineers have managed to fit a QWERTY keyboard onto the handset's slim frame.
 </data_item>
</magento_api>
```

```

</short_description>
<name>BlackBerry 8100 Pearl</name>
<meta_title>BlackBerry 8100 Pearl</meta_title>
<meta_description>BlackBerry 8100 Pearl sports a large 240 x 260 screen that
supports over 65,000 colors-- plenty of real estate to view your e-mails, Web browser
content, messaging sessions, and attachments.</meta_description>
<regular_price_with_tax>349.99</regular_price_with_tax>
<regular_price_without_tax>349.99</regular_price_without_tax>
<final_price_with_tax>349.99</final_price_with_tax>
<final_price_without_tax>349.99</final_price_without_tax>
<is_saleable>1</is_saleable>
<image_url>http://magentohost/imageulr/blackberry.jpg</image_url>
</data_item>
</magento_api>

```

### **Additional Information**

You can define the limit of items returned in the response by passing the limit parameter. By default, 10 items are returned and the maximum number is 100 items. You can also define the page number by passing the page parameter. Example:

<http://magentohost/api/rest/products?page=2&limit=20>

Authorization header will be required for Admin and Customer user types. The following parameters must be provided in the Authorization header for the call:

- oauth\_consumer\_key - the Consumer Key value provided after the registration of the application.
- oauth\_nonce - a random value, uniquely generated by the application.
- oauth\_signature\_method - name of the signature method used to sign the request. Can have one of the following values: HMAC-SHA1, RSA-SHA1, and PLAINTEXT.
- oauth\_signature - a generated value (signature).
- oauth\_timestamp - a positive integer, expressed in the number of seconds since January 1, 1970 00:00:00 GMT.
- oauth\_token - the oauth\_token value (Request Token).
- oauth\_version - OAuth version.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Authentication

[Edit this page on GitHub](#) 

## Introduction

### Authentication

OAuth Configuration

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

### Permission Settings

### Resources

## Response Formats

## Testing REST Resources

## Introduction

Using OAuth

Why Do You Need OAuth?

OAuth Definitions

OAuth Process

Registering an Application

Authentication Endpoints

Getting an Unauthorized Request Token

User Authorization

Getting an Access Token

OAuth Error Codes

PHP Examples

Retrieve the list of products for Admin user with OAuth authentication

Retrieve the list of products for Customer user with OAuth authentication

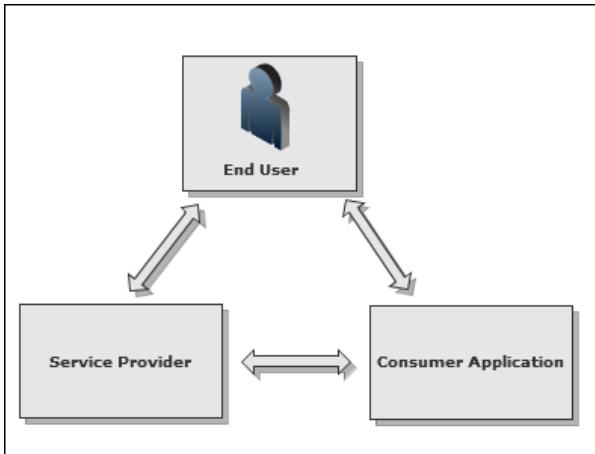
## Introduction

In most cases, the third-party application must be authenticated to use the Magento API. But users never reveal

their credentials to the application to preserve their privacy. So, the question is as follows: how is your application going to authenticate users if it does not know user credentials. OAuth is the solution.

Magento authentication is based on OAuth, an open standard for secure API authentication. OAuth is a token-passing mechanism that allows users to control which applications have access to their data without revealing their passwords or other credentials.

The OAuth concept lies in three basic elements that can be easily described in the following picture:



To learn more about OAuth, you can visit the official [OAuth](#) site.

## Using OAuth

The current API supports OAuth 1.0a.

The OAuth authentication works by asking the user to authorize their application. When the user authorizes the application, the application can access that user protected resources by using an access token. This token will be used in each further request. Access tokens are long-lived and will not expire unless the user revokes access to the application.

OAuth is completely invisible for the site visitors.

## Why Do You Need OAuth?

Magento uses OAuth to allow access to its data. You need to use OAuth if you want to use any of the following Magento APIs:

- Products
- Inventory
- Orders
- Customers
- Customer Addresses
- Categories

and a lot more

## OAuth Definitions

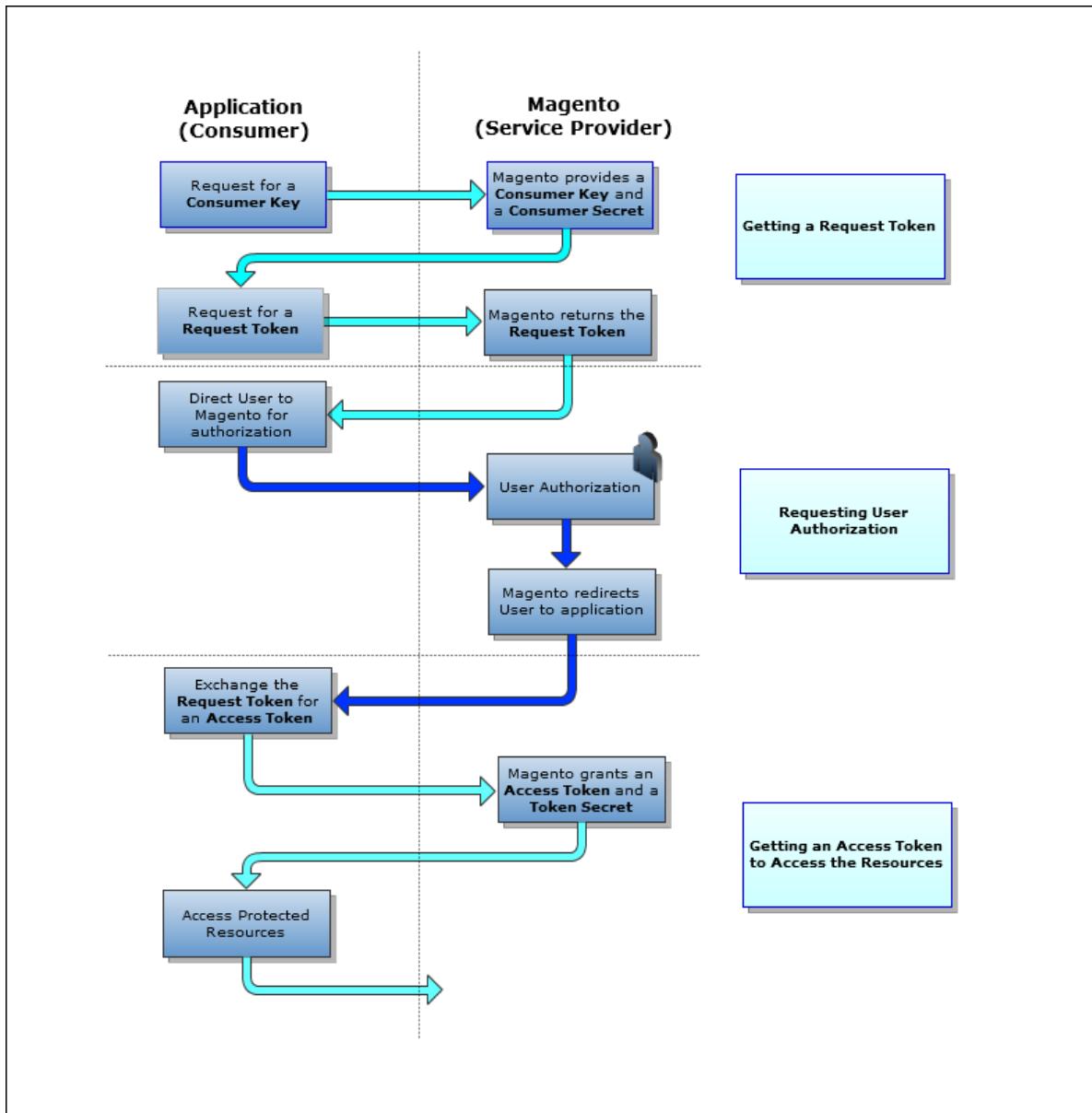
There are some definitions you need to get familiar with before you start using OAuth. These are as follows:

- **User** - A customer who has an account with Magento and can use the services via the Magento API.
- **Consumer** - A third-party application that uses OAuth to access the Magento API. This application must be registered in the Magento system to receive the Consumer Key and Consumer Secret.
- **Consumer Key** - A value used by the Consumer to identify itself with Magento.
- **Consumer Secret** - A secret used by the Consumer to guarantee the ownership of the Consumer Key. This value is not passed in requests.
- **Request Token** - A value used by the Consumer to obtain authorization from the User (when needed). The Request Token is exchanged for an Access Token when permission is granted.
- **Access Token** - A value used by the Consumer to call Magento APIs on behalf of the User.

## OAuth Process

The OAuth process consists of several steps:

- Getting an Unauthorized Request Token.
- Requesting user authorization.
- Getting an Access Token by exchanging the Request Token for it.



The application that requires access to data is known as the Consumer and Magento is the Service Provider.

## Registering an Application

Before starting to make API requests, you need to register the application. After the registration, you will receive the Consumer Key that will identify you in Magento. Also, you will receive a Consumer Secret. This secret will be used when requesting for a Request Token.

You can register your application by selecting **System > Web Services > REST - OAuth Consumers** and clicking **Add New** in the Admin Panel.

When registering the application, you also need to define the callback URL, to which the user will be redirected after he/she successfully authorizes your application.

## Authentication Endpoints

The authentication endpoints include the following ones:

- **/oauth/initiate** - this endpoint is used for retrieving the Request Token.
- **/oauth/authorize** - this endpoint is used for user authorization (Customer).
- **/admin/oauth\_authorize** - this endpoint is used for user authorization (Admin).
- **/oauth/token** - this endpoint is used for retrieving the Access Token.

Also, the simple form can be used for authentication. To use a simple form, add the /simple endpoint to the authentication endpoint. For example: /oauth/authorize/simple

### Getting an Unauthorized Request Token

The first step to authenticate the user is to retrieve a Request Token from Magento. This is a temporary token that will be exchanged for the Access Token.

|                         |                                                                                                                                |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>Endpoint:</b>        | /oauth/initiate                                                                                                                |
| <b>Description:</b>     | The first step of authentication. Allows you to obtain the Request Token used for the rest of the authentication process.      |
| <b>Method:</b>          | POST                                                                                                                           |
| <b>Returns:</b>         | Request Token                                                                                                                  |
| <b>Sample Response:</b> | oauth_token=4cqw0r7vo0s5goyyqnjb72sqj3vxwr0h&oauth_token_secret=rig3x3j5a9z5j6d4ubjwyf9f1l21itrr&oauth_callback_confirmed=true |

The following request parameters should be present in the Authorization header:

- oauth\_callback - an URI to which the Service Provider will redirect the resource owner (user) after the authorization is complete.
- oauth\_consumer\_key - the Consumer Key value, retrieved after the registration of the application.
- oauth\_nonce - a random value, uniquely generated by the application.
- oauth\_signature\_method - name of the signature method used to sign the request. Can have one of the following values: HMAC-SHA1, RSA-SHA1, and PLAINTEXT.
- oauth\_signature - a generated value (signature).
- oauth\_timestamp - a positive integer, expressed in the number of seconds since January 1, 1970 00:00:00 GMT.
- oauth\_version - OAuth version.

## User Authorization

The second step is to request user authorization. After receiving the Request Token from Magento, the application provides an authorization page to the user. The only required parameter for this step is the Request Token (oauth\_token value) received from the previous step. The endpoint is followed by an oauth\_token parameter with the value set to the oauth\_token value.

After this, the user is asked to enter their credentials and authorize. When the user is granted the access, he/she is redirected to the URL specified in the oauth\_callback parameter. This URL is followed by two parameters:

- oauth\_token - the Request Token value.
- oauth\_verifier - a verification code that is tied to the Request Token.

|                         |                                                                                                                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Endpoint:</b>        | /oauth/authorize                                                                                                                                 |
| <b>Description:</b>     | The second step of authentication. Without the user authorization in this step, it is impossible for your application to obtain an Access Token. |
| <b>Method:</b>          | GET                                                                                                                                              |
| <b>Sample Response:</b> | /callback?<br>oauth_token=tz2kmxyf3lagl3o95xno9ia15k6mpt3&oauth_verifier=cbwwh03alr5huiz5c76wi4l21zf05eb0                                        |

## Getting an Access Token

The final third authentication step. After the application access is authorized, the application needs to exchange the Request Token for an Access Token. For this step, you will need the Request Token (the oauth\_token and oauth\_token\_secret values) and the oauth\_verifier value from the previous step.

|                         |                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------|
| <b>Endpoint:</b>        | /oauth/token                                                                                      |
| <b>Description:</b>     | The third step of authentication. Getting an Access Token.                                        |
| <b>Method:</b>          | POST                                                                                              |
| <b>Returns:</b>         | An access token and the corresponding access token secret, URL-encoded.                           |
| <b>Sample Response:</b> | oauth_token=0lnuajnuzeei2o8xcddii5us77xnb6v0&oauth_token_secret=1c6d2hycnir5ygf39fyccs6zhtaagx8pd |

The following components should be present in the Authorization header:

- oauth\_consumer\_key - the Consumer Key value provided after the registration of the application.
- oauth\_nonce - a random value, uniquely generated by the application.
- oauth\_signature\_method - name of the signature method used to sign the request. Can have one of the following values: HMAC-SHA1, RSA-SHA1, and PLAINTEXT.
- oauth\_signature - a generated value (signature).
- oauth\_timestamp - a positive integer, expressed in the number of seconds since January 1, 1970 00:00:00 GMT.

- oauth\_token - the oauth\_token value (Request Token) received from the previous steps.
- oauth\_verifier - the verification code that is tied to the Request Token.
- oauth\_version - OAuth version.

The response will contain the following response parameters:

- oauth\_token - the Access Token that provides access to protected resources.
- oauth\_token\_secret - the secret that is associated with the Access Token.

## OAuth Error Codes

When the third-party application performs invalid requests to Magento, the following errors related to OAuth can occur:

| HTTP Code | Error Code | Text Representation       | Description                                                                                                                                                                 |
|-----------|------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 400       | 1          | version_rejected          | This error is used when the oauth_version parameter does not correspond to the "1.0a" value.                                                                                |
| 400       | 2          | parameter_absent          | This error is used there is no required parameter in the request. The name of the missing parameter is specified additionally in the response.                              |
| 400       | 3          | parameter_rejected        | This error is used when the type of the parameter or its value does not meet the protocol requirements (e.g., array is passed instead of the string).                       |
| 400       | 4          | timestamp_refused         | This error is used if there is incorrect value of the timestamp in the oauth_timestamp parameter.                                                                           |
| 401       | 5          | nonce_used                | This error is used if the nonce-timestamp combination has already been used.                                                                                                |
| 400       | 6          | signature_method_rejected | This error is used for unsupported signature method. The following methods are supported: HMAC-SHA1, RSA-SHA1, and PLAINTEXT.                                               |
| 401       | 7          | signature_invalid         | This error is used if the signature is invalid.                                                                                                                             |
| 401       | 8          | consumer_key_rejected     | This error is used if the Consumer Key has incorrect length or does not exist.                                                                                              |
| 401       | 9          | token_used                | This error is used if there is an attempt of authorization of an already authorized token or an attempt to exchange a not temporary token for a permanent one.              |
| 401       | 10         | token_expired             | This error is used if the temporary token has expired. At the moment, the mechanism of expiration of temporary tokens is not implemented and the current error is not used. |
| 401       | 11         | token_revoked             | This error is used if the token is revoked by the user who authorized it.                                                                                                   |
| 401       | 12         | token_rejected            | This error is used if the token is not valid, or does not exist, or is not valid for                                                                                        |

|     |    |                  |                                                                                 |
|-----|----|------------------|---------------------------------------------------------------------------------|
|     |    |                  | using in the current type of request.                                           |
| 401 | 13 | verifier_invalid | This error is used if the confirmation string does not correspond to the token. |

## PHP Examples

Retrieve the list of products for Admin user with OAuth authentication

```
<?php
/**
 * Example of retrieving the products list using Admin account via Magento
 * REST API. OAuth authorization is used
 * Preconditions:
 * 1. Install php oauth extension
 * 2. If you were authorized as a Customer before this step, clear browser
 * cookies for 'yourhost'
 * 3. Create at least one product in Magento
 * 4. Configure resource permissions for Admin REST user for retrieving all
 * product data for Admin
 * 5. Create a Consumer
 */
// $callbackUrl is a path to your file with OAuth authentication example for
the Admin user
$callbackUrl = "http://yourhost/oauth_admin.php";
$temporaryCredentialsRequestUrl = "http://yourhost/oauth/initiate?
oauth_callback=" . urlencode($callbackUrl);
$adminAuthorizationUrl = 'http://yourhost/admin/oAuth_authorize';
$accessTokenRequestUrl = 'http://yourhost/oauth/token';
$apiUrl = 'http://yourhost/api/rest';
$consumerKey = 'yourconsumerkey';
$consumerSecret = 'yourconsumersecret';

session_start();
if (!isset($_GET['oauth_token']) && !isset($_SESSION['state']) &&
$_SESSION['state'] == 1) {
 $_SESSION['state'] = 0;
}
try {
 $authType = ($_SESSION['state'] == 2) ? OAUTH_AUTH_TYPEORIZATION :
OAUTH_AUTH_TYPE_URI;
 $oauthClient = new OAuth($consumerKey, $consumerSecret,
OAUTH_SIG_METHOD_HMACSHA1, $authType);
 $oauthClient->enableDebug();

 if (!isset($_GET['oauth_token']) && !$SESSION['state']) {
 $requestToken = $oauthClient-
>getRequestToken($temporaryCredentialsRequestUrl);
```

```

$_SESSION['secret'] = $requestToken['oauth_token_secret'];
$_SESSION['state'] = 1;
header('Location: ' . $adminAuthorizationUrl . '?oauth_token=' .
$requestToken['oauth_token']);
exit;
} else if ($_SESSION['state'] == 1) {
 $oAuthClient->setToken($_GET['oauth_token'], $_SESSION['secret']);
 $accessToken = $oAuthClient->getAccessToken($accessTokenRequestUrl);
 $_SESSION['state'] = 2;
 $_SESSION['token'] = $accessToken['oauth_token'];
 $_SESSION['secret'] = $accessToken['oauth_token_secret'];
 header('Location: ' . $callbackUrl);
 exit;
} else {
 $oAuthClient->setToken($_SESSION['token'], $_SESSION['secret']);

 $resourceUrl = "$apiUrl/products";
 $oAuthClient->fetch($resourceUrl, array(), 'GET', array('Content-
Type' => 'application/json'));
 $productsList = json_decode($oAuthClient->getLastResponse());
 print_r($productsList);
}
} catch (OAuthException $e) {
 print_r($e->getMessage());
 echo "
";
 print_r($e->lastResponse);
}

```

## Retrieve the list of products for Customer user with OAuth authentication

```

<?php
/**
 * Example of retrieving the products list using Customer account via Magento REST
 * API. OAuth authorization is used
 * Preconditions:
 * 1. Install php oauth extension
 * 2. If you were authorized as an Admin before this step, clear browser cookies for
 * 'yourhost'
 * 3. Create at least one product in Magento and enable it for viewing in the frontend
 * 4. Configure resource permissions for Customer REST user for retrieving all product
 * data for Customer
 * 5. Create a Consumer
 */
// $callbackUrl is a path to your file with OAuth authentication example for the
Customer user
$callbackUrl = "http://yourhost/oauth_customer.php";
$temporaryCredentialsRequestUrl = "http://yourhost/oauth/initiate?oauth_callback=" .
urlencode($callbackUrl);
$customerAuthorizationUrl = 'http://yourhost/oauth/authorize';

```

```

$accessTokenRequestUrl = 'http://yourhost/oauth/token';
$apiUrl = 'http://yourhost/api/rest';
$consumerKey = 'yourconsumerkey';
$consumerSecret = 'yourconsumersecret';

session_start();
if (!isset($_GET['oauth_token']) && isset($_SESSION['state']) && $_SESSION['state'] == 1) {
 $_SESSION['state'] = 0;
}
try {
 $authType = ($_SESSION['state'] == 2) ? OAUTH_AUTH_TYPE_AUTHORIZATION : OAUTH_AUTH_TYPE_URI;
 $oauthClient = new OAuth($consumerKey, $consumerSecret, OAUTH_SIG_METHOD_HMACSHA1, $authType);
 $oauthClient->enableDebug();

 if (!isset($_GET['oauth_token']) && !$_SESSION['state']) {
 $requestToken = $oauthClient->getRequestToken($temporaryCredentialsRequestUrl);
 $_SESSION['secret'] = $requestToken['oauth_token_secret'];
 $_SESSION['state'] = 1;
 header('Location: ' . $customerAuthorizationUrl . '?oauth_token=' . $requestToken['oauth_token']);
 exit;
 } else if ($_SESSION['state'] == 1) {
 $oauthClient->setToken($_GET['oauth_token'], $_SESSION['secret']);
 $accessToken = $oauthClient->getAccessToken($accessTokenRequestUrl);
 $_SESSION['state'] = 2;
 $_SESSION['token'] = $accessToken['oauth_token'];
 $_SESSION['secret'] = $accessToken['oauth_token_secret'];
 header('Location: ' . $callbackUrl);
 exit;
 } else {
 $oauthClient->setToken($_SESSION['token'], $_SESSION['secret']);

 $resourceUrl = "$apiUrl/products";
 $oauthClient->fetch($resourceUrl, array(), 'GET', array('Content-Type' => 'application/json'));
 $productsList = json_decode($oauthClient->getLastResponse());
 print_r($productsList);
 }
} catch (OAuthException $e) {
 print_r($e->getMessage());
 echo "
";
 print_r($e->lastResponse);
}

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# OAuth Configuration

[Edit this page on GitHub](#) 

## Introduction

## Authentication

[OAuth Configuration](#)

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings

## Resources

## Response Formats

## Testing REST Resources

---

### Working with Consumers

[Adding a New Consumer](#)

[Editing an Existing Consumer](#)

[Deleting an Existing Consumer](#)

[Searching for a Consumer](#)

### Working with Tokens (Admin Panel)

[Viewing Authorized Tokens](#)

[Viewing Applications](#)

[Enabling Tokens](#)

[Revoking Tokens](#)

[Deleting Tokens](#)

### Working with Tokens (Frontend)

[Viewing Applications](#)

[Enabling Tokens](#)

[Disabling Tokens](#)

[Deleting Tokens](#)

### Working with Email Templates

[Setting Up the Default Email Template](#)

[Creating a New Email Template](#)

### Cleanup Configuration

Before starting to use OAuth, you need to perform several steps in the Magento Admin Panel. These steps allow you to enable the OAuth functionality for further actions.

# Working with Consumers

## Adding a New Consumer

First, you need to create a Consumer in the Admin Panel. Creating a new consumer means registering the application. To do this, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - OAuth Consumers**.
2. On the OAuth Consumers page, click **Add New** in the top right corner to add a new consumer.
3. The New Consumer page opens. The **Key** and **Secret** fields are filled automatically and cannot be edited. These values are generated automatically and will be used to identify the Consumer in Magento.

The screenshot shows the 'New Consumer' form in the Magento Admin Panel. The 'Consumer Information' section contains the following fields:

- Name: My Consumer
- Key: z83i5q63wgdwl9wtl2euhwdd9jfr2s2
- Secret: hamnk1ca1vwlu037l38n0uoxaspi61ze
- Callback URL: (empty)
- Rejected Callback URL: (empty)

At the top right of the form, there are four buttons: 'Back', 'Reset', 'Save', and 'Save and Continue Edit'.

4. Fill in the following fields:
  - **Name**: Enter the name of the application to be registered. This field is required.
  - **Callback URL**: Enter the URL address to which the Consumer will be redirected after the authorization is passed successfully. This URL address implies the path to the application. This field is optional.
  - **Rejected Callback URL**: Enter the URL address to which the user will be redirected if he/she rejects authorization. This field is optional.
5. Click **Save** in the top right corner to save the created Consumer.

## Editing an Existing Consumer

To edit an existing consumer, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - OAuth Consumers**.
2. The OAuth Consumers page opens. In the grid, select the consumer to be edited and click it.
3. The Edit Consumer page opens. On this page, you can edit the following fields:
  - **Name**: Enter a new name for the application.
  - **Callback URL**: Enter a new URL address to which the user will be redirected after successful authorization. This URL address implies the path to the application.
  - **Rejected Callback URL**: Enter another URL address to which the user will be redirected after he/she rejects authorization proceeding.

The **Key** and **Secret** fields cannot be edited.

4. Click **Save** in the top right corner to save changes.

## Deleting an Existing Consumer

To delete the required consumer, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - OAuth Consumers**.
2. The OAuth Consumers page opens. In the grid, select the consumer to be deleted and click it.
3. The Edit Consumer page opens. Click **Delete** in the top right corner to delete the selected consumer.

## Searching for a Consumer

You can search for a required consumer by several parameters: ID, consumer name, and date of creation.

To search for a consumer, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - OAuth Consumers**.
2. The OAuth Consumers page opens. The list of consumers is displayed in a grid with the following fields: ID, Consumer Name, and Created At.
3. In the search field below the column header in a grid, enter the required value by which the search will be performed. Click **Search** in the top right corner.

## Working with Tokens (Admin Panel)

### Viewing Authorized Tokens

To view authorized tokens in the Admin panel, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - OAuth Authorized Tokens**.
2. The Authorized OAuth Tokens page opens. In the grid, the list of all authorized tokens is displayed.
3. Tokens are displayed in the grid with the following columns: ID, Application Name (name of consumer for which the token is created), User Type (type of the user, Customer or Admin), User ID, and the Revoked status.

From the Authorized OAuth Tokens page, you can enable, revoke, or delete the required token.

### Viewing Applications

To view the list of applications, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - My Apps**.
2. The My Applications page opens. Registered applications are displayed in a grid with the following columns: ID, Application Name, and Revoked.

## Enabling Tokens

If a token is revoked (the Yes value in the Revoked column on the Authorized OAuth Tokens page), you can enable it. To do this, perform the following steps:

1. In the Authorized OAuth Tokens grid, select the token with the **Revoked** status set to Yes and select the checkbox next to it.

You can select more than one token with the **Revoked** status and enable all of them by using the mass action.

2. In the **Actions** drop-down list, select the **Enable** option and click **Submit**.

3. The required token is enabled.

## Revoking Tokens

If a token is enabled (the No value in the Revoked column), you can revoke it. To do this, perform the following steps:

1. In the Authorized OAuth Tokens grid, select the token with the **Revoked** status set to No and select the checkbox next to it.

You can select more than one token with the **Revoked** status set to No and revoke all of them by using the mass action.

2. In the **Actions** drop-down list, select the **Revoke** option and click **Submit**.

3. The required token is revoked.

## Deleting Tokens

To delete the required token, perform the following steps:

1. In the Authorized OAuth Tokens grid, select the token to be deleted and select the checkbox next to it.

You can select more than one token and delete all of them by using the mass action.

2. In the **Actions** drop-down list, select the **Delete** option and click **Submit**.

3. The required token is deleted.

## Working with Tokens (Frontend)

### Viewing Applications

To view the authorized applications from the frontend, perform the following steps:

1. On the frontend, click **My Account** and then select the **My Applications** tab on the left.

2. On the My Applications page, the list of registered applications will be displayed.

| My Account             | My Applications                                  |                                                 |
|------------------------|--------------------------------------------------|-------------------------------------------------|
|                        | APP NAME                                         | STATUS                                          |
| Account Dashboard      | My App                                           | Enabled                                         |
| Account Information    | <a href="#">Disable</a>   <a href="#">Delete</a> |                                                 |
| Address Book           |                                                  |                                                 |
| My Orders              |                                                  |                                                 |
| Billing Agreements     |                                                  |                                                 |
| Recurring Profiles     |                                                  |                                                 |
| My Product Reviews     |                                                  |                                                 |
| My Tags                |                                                  |                                                 |
| My Wishlist            |                                                  |                                                 |
| <b>My Applications</b> |                                                  |                                                 |
| 1 Item(s)              |                                                  | Show <input type="button" value="10"/> per page |
|                        |                                                  |                                                 |
|                        |                                                  |                                                 |
|                        |                                                  |                                                 |
| 1 Item(s)              |                                                  | Show <input type="button" value="10"/> per page |
|                        |                                                  |                                                 |
|                        |                                                  |                                                 |
|                        |                                                  |                                                 |

From this page, you can enable, revoke, or delete the required token.

## Enabling Tokens

1. In the list of consumers, select the consumer to be enabled.

If the token is revoked, there will be the **Disabled** status next to it.

2. Click **Enable** next to the consumer name.
3. The token is enabled.

## Disabling Tokens

1. In the list of consumers, select the consumer to be disabled.

If the token is enabled, there will be the **Enabled** status next to it.

2. Click **Disable** next to the consumer name.
3. The token is disabled.

## Deleting Tokens

1. In the list of consumers, select the consumer to be deleted.
2. Click **Delete** next to the consumer name.
3. The token is deleted.

## Working with Email Templates

### Setting Up the Default Email Template

You can set the email template that will be used for user notification if the token status changes. Also, you can set different email templates for different store views. For example, you have two store views: English and German. Magento allows you to set one email template for the English store view and another one for the German store view.

To set the email template, perform the following steps:

1. On the Admin Panel menu, select **System > Configuration**.
2. Select **Services > OAuth** on the left.

3. In the **Email** panel, from the **Token Status Change Email Template** drop-down list, select the required template and click **Save Config** in the top right corner.
4. The template is saved.

## Creating a New Email Template

You can also create your own email template that will be used for user notification if the token status changes. To create a new template, perform the following steps:

1. On the Admin Panel menu, select **System > Transactional Emails**.
2. The Transactional Emails page opens. Click **Add New Template** in the top right corner.
3. The New Email template page opens. In the Load default template panel, in the **Template** drop-down list, select the **Token Status Change** option.
4. Specify the **Locale** option and click **Load Template**.
5. In the Template Information panel, the template data is loaded. You can specify your own template name, subject, and content.
6. When the email template is created, click **Save Template** in the top right corner.
7. Set the newly created template as it was described above.

## Cleanup Configuration

You can configure the cleanup functionality for temporary tokens. These tokens can be deleted after a certain period of time or after a certain number of OAuth requests.

To configure cleanup, perform the following steps:

1. On the Admin Panel menu, select **System > Configuration**.
2. Select **Services > OAuth** on the left.

Current Configuration Scope: [?](#)  
Default Config

[Manage Stores](#)

**Configuration**

- ▶ **GENERAL**
  - General
  - Web
  - Design
  - Currency Setup
  - Store Email Addresses
  - Contacts
  - Reports
  - Content Management

**OAuth**

**Email**

Token Status Change Email Template  [STORE VIEW]

**Cleanup Settings**

Cleanup Probability  [GLOBAL]  
▲ Integer. Launch cleanup in X OAuth requests. 0 (not recommended) - to disable cleanup.

Expiration Period  [GLOBAL]  
▲ Cleanup entries older than X minutes.

[Save Config](#)

3. In the **Cleanup Statistics** panel, you can set the following values:

- **Cleanup Probability:** Define the threshold of OAuth requests after which the cleanup will be performed. Only temporary credentials will be removed. Enter 0 to disable the cleanup.
- **Expiration Period:** Define the period (in minutes) on the expiry of which entries will be deleted from the database.

4. Click **Save Config** in the top right corner to save changes.

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Common HTTP Status Codes

[Edit this page on GitHub](#) 

### Introduction

[HTTP Status Codes](#)

### Authentication

[Error Messages](#)

### Common HTTP Status Codes

### HTTP Methods

HTTP status codes are an essential part of the REST concept. You can get familiar with all of them on [Wikipedia](#).

### GET Filters

The Magento API attempts to return appropriate HTTP status codes for all requests. Any information is returned in the form of a standard HTTP response with an HTTP status code describing the error and the body message.

### Permission Settings

### Resources

### Response Formats

### Testing REST Resources

## HTTP Status Codes

The following table contains possible common HTTP status codes:

| Status Code      | Message                                                                                                                                                                                                                                                                                                                                                    |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 200 OK           | -                                                                                                                                                                                                                                                                                                                                                          |
| 201 Created      | Resource was partially created                                                                                                                                                                                                                                                                                                                             |
| 207 Multi-Status | -                                                                                                                                                                                                                                                                                                                                                          |
| 400 Bad Request  | Resource data pre-validation error.<br>Resource data invalid.<br>Resource unknown error.<br>The request data is invalid.<br>Resource collection paging error.<br>The paging limit exceeds the allowed number.<br>Resource collection ordering error.<br>Resource collection filtering error.<br>Resource collection including additional attributes error. |
| 403 Forbidden    | Access denied.                                                                                                                                                                                                                                                                                                                                             |
| 404 Not Found    | Resource not found.                                                                                                                                                                                                                                                                                                                                        |

|                        |                                                                           |
|------------------------|---------------------------------------------------------------------------|
| 405 Method Not Allowed | Resource does not support method.<br>Resource method not implemented yet. |
| 500 Internal Error     | Unhandled simple errors.<br>Resource internal error.                      |

## Error Messages

When the Magento API returns an error message, it returns it in your requested format. For example, an error in the XML format might look like the following:

```
<?xml version="1.0"?>
<magento_api>
 <messages>
 <error>
 <data_item>
 <code>404</code>
 <message>Resource not found.</message>
 </data_item>
 </error>
 </messages>
</magento_api>
```

An error in the JSON format might look like the following:

```
{"messages":{"error":[{"code":404,"message":"Resource not found."}]}}
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# HTTP Methods

[Edit this page on GitHub](#) 

## Introduction

## Authentication

## Common HTTP Status Codes

## HTTP Methods

### GET Filters

### Permission Settings

### Resources

### Response Formats

### Testing REST Resources

#### GET

#### POST and PUT

#### DELETE

Accessing API is performed via HTTP. When you enter a URL into a web browser address bar, the browser performs an HTTP GET request to the URL. This usually returns a web page in the form of an HTTP response that the browser displays. But the GET method is one of several HTTP request methods. Magento REST API uses the four main HTTP methods: GET, POST, PUT, and DELETE. The most widespread methods are GET and POST. The other methods are less known but they became widely known due to the popularity of REST web services. An important concept of the REST architecture is that different HTTP request methods perform different actions when applied to the same URL.

For example:

```
GET https://magentohost/rest/customers/123
```

will retrieve information about the specified customer;

```
DELETE https://magentohost/rest/customers/123
```

will delete the specified customer.

## GET

### Retrieving Resources with the HTTP GET Method

The HTTP GET method is defined in section 9.3 of the [RFC2616](#) document:

The GET method means retrieve whatever information (in the form of an entity) is identified by the Request-URI. If the Request-URI refers to a data-producing process, it is the produced data which shall be returned as the entity in the response and not the source text of the process, unless that text happens to be the output of the process.

You can retrieve a representation of a resource by getting its URL.

# POST and PUT

## **Creating or Updating Resources with the HTTP POST and PUT Methods**

The POST method is defined in section 9.5 of the [RFC2616](#) document:

The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. POST is designed to allow a uniform method to cover the following functions:

- Annotation of existing resources;
- Posting a message to a bulletin board, newsgroup, mailing list, or similar group of articles;
- Providing a block of data, such as the result of submitting a form, to a data-handling process;
- Extending a database through an append operation.

The PUT method is defined in section 9.6 of the [RFC2616](#) document:

The PUT method requests that the enclosed entity be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource, the enclosed entity SHOULD be considered as a modified version of the one residing on the origin server.

Creating or updating a resource involves performing an HTTP POST or HTTP PUT to a resource URL.

# DELETE

## **Deleting Resources with the HTTP DELETE Method**

The DELETE method is defined in section 9.7 of the [RFC2616](#) document:

The DELETE method requests that the origin server delete the resource identified by the Request-URI. This method MAY be overridden by human intervention (or other means) on the origin server.

Deleting a resource is performed by means of making an HTTP DELETE request to the resource URL.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# GET Filters

[Edit this page on GitHub](#) 

*JSON responses on this page contributed by Tim Reynolds*

## Introduction

## Authentication

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings

## Resources

## Response Formats

## Testing REST Resources

Some requests use GET parameters in the URL. These are as follows:

- **filter** - specifies the filters for returned data
- **page** - specifies the page number which items will be returned
  - e.g., <http://magentohost/api/rest/products?page=1>
- **order, dir** - specifies the sort order of returned items and the order direction: 'asc' - returns items in the ascending order; 'dsc' - returns items in the descending order.
  - e.g., <http://magentohost/api/rest/products?order=name&dir=dsc>
  - e.g., <http://magentohost/api/rest/products?order=name&dir=asc>
- **limit** - limits the number of returned items in the response. Note that by default, 10 items are returned in the response. The maximum number is 100 items.
  - e.g., <http://magentohost/api/rest/products?limit=2>
- **neq** - "not equal to" - returns items with the specified attribute that is not equal to the defined value
  - e.g., [http://magentohost/api/rest/products?filter\[1\]\[attribute\]=entity\\_id&filter\[1\]\[neq\]=3](http://magentohost/api/rest/products?filter[1][attribute]=entity_id&filter[1][neq]=3)
- **in** - "equals any of" - returns items that are equal to the item(s) with the specified attribute(s)
  - e.g., [http://magentohost/api/rest/products?filter\[1\]\[attribute\]=entity\\_id&filter\[1\]\[in\]=3](http://magentohost/api/rest/products?filter[1][attribute]=entity_id&filter[1][in]=3)
- **nin** - "not equals any of" - returns items excluding the item with the specified attribute
  - e.g., [http://magentohost/api/rest/products?filter\[1\]\[attribute\]=entity\\_id&filter\[1\]\[nin\]=3](http://magentohost/api/rest/products?filter[1][attribute]=entity_id&filter[1][nin]=3)
- **gt** - "greater than" - returns items with the specified attribute that is greater than the defined value
  - e.g., [http://magentohost/api/rest/products?filter\[1\]\[attribute\]=entity\\_id&filter\[1\]\[gt\]=3](http://magentohost/api/rest/products?filter[1][attribute]=entity_id&filter[1][gt]=3)
  - e.g., [http://magentohost/api/rest/products?filter\[1\]\[attribute\]=price&filter\[1\]\[gt\]=300](http://magentohost/api/rest/products?filter[1][attribute]=price&filter[1][gt]=300)
- **lt** - "less than" - returns items with the specified attribute that is less than the defined value
  - e.g., [http://magentohost/api/rest/products?filter\[1\]\[attribute\]=entity\\_id&filter\[1\]\[lt\]=4](http://magentohost/api/rest/products?filter[1][attribute]=entity_id&filter[1][lt]=4)
- **from, to** - specifies the range of attributes according to which items will be returned
  - e.g., [http://magentohost/api/rest/products?filter\[1\]\[attribute\]=entity\\_id&filter\[1\]\[from\]=1&filter\[1\]\[to\]=3](http://magentohost/api/rest/products?filter[1][attribute]=entity_id&filter[1][from]=1&filter[1][to]=3)

- e.g., [http://magentohost/api/rest/products?filter\[1\]\[attribute\]=price&filter\[1\]\[from\]=150&filter\[1\]\[to\]=350](http://magentohost/api/rest/products?filter[1][attribute]=price&filter[1][from]=150&filter[1][to]=350)

If the attribute value consists of several words separated by a whitespace, the '%20' sign is used:

- e.g., [http://magentohost/api/rest/products?filter\[1\]\[attribute\]=name&filter\[1\]\[in\]=BlackBerry%208100%20Pearl](http://magentohost/api/rest/products?filter[1][attribute]=name&filter[1][in]=BlackBerry%208100%20Pearl)

For example, to filter products with the description equal to simple01:

[http://magentohost/api/rest/products/?order=entity\\_id&filter\[0\]\[attribute\]=description&filter\[0\]\[in\]\[0\]=simple01](http://magentohost/api/rest/products/?order=entity_id&filter[0][attribute]=description&filter[0][in][0]=simple01)

To filter customers by email address:

[http://magentohost/api/rest/customers?filter\[1\]\[attribute\]=email&filter\[1\]\[in\]\[0\]=ryan@test.com](http://magentohost/api/rest/customers?filter[1][attribute]=email&filter[1][in][0]=ryan@test.com)

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Permission Settings

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ^

Roles Configuration

Attributes Configuration

Attributes Description

## Resources ▾

## Response Formats

## Testing REST Resources

## Introduction

Authorization

Access Levels

Access Control List (ACL)

ACL Overview

Understanding ACL

ACL Structure

Read/Write Permissions

Setting Up ACL

## Introduction

After the authentication is complete successfully, the Access Token is received and will be used in every API call. This key allows identifying the client that accesses the API. With the help of this key, the following information about the user can be retrieved:

- Type of the API user
- User ID (can be Admin ID or Customer ID)

## Authorization

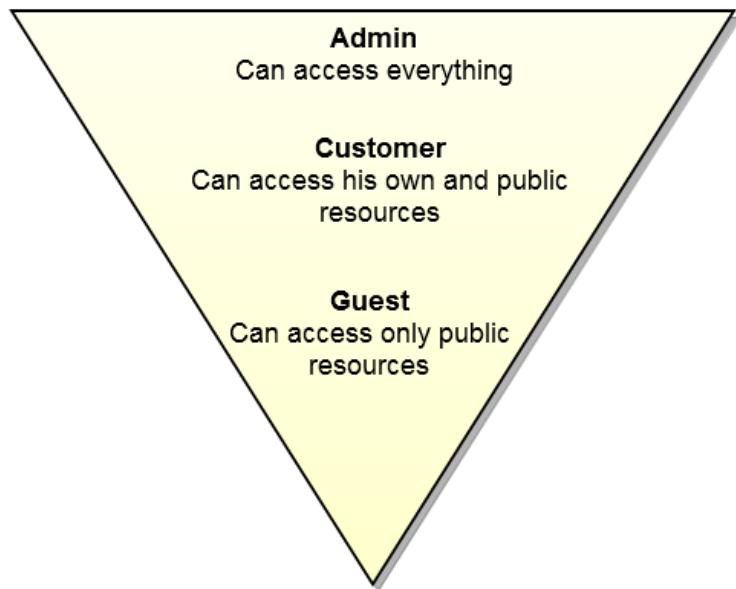
## Access Levels

There is a three-level authorization approach in Magento REST API. These three levels are as follows:

- Guest
- Customer
- Admin

The following graphic describes the default rights for each access level with each level obtaining more rights

up to Admin who has access to everything.



Each user type will be described below.

Magento grants permissions for the following three types of users:

#### **Guest**

Guest can be a type of application that does not require authentication. This application has access only to public resources.

#### **Customer**

Customer can be a registered and logged in user. This type of user can have access only to its own resources as well as to public resources.

#### **Admin**

Admin can be the store owner. This type has full set of permissions.

Understanding of access levels is the basis of the ACL work.

## Access Control List (ACL)

### **ACL Overview**

Every user has a specific role and purpose. To accomplish their goals, each user must be able to access certain resources and perform specific actions. Allowing users to access the resources without any limits can compromise Magento security.

The Access Control List (ACL) is a set of permissions (access rights) that particular users have for certain resources. When a user wants to perform a specific action with a resource (for example, update the customer information), Magento checks the permission for this combination of user, resource, and action. If the action is allowed, the user can proceed. Otherwise, the action is denied.

### **Understanding ACL**

Access control lists include two main things: a subject and an object. Usually, the subject is the user who wants to use the resource. The object is the resource that a certain user wants to have access to. So, ACL is used to decide when the subject can have access to object.

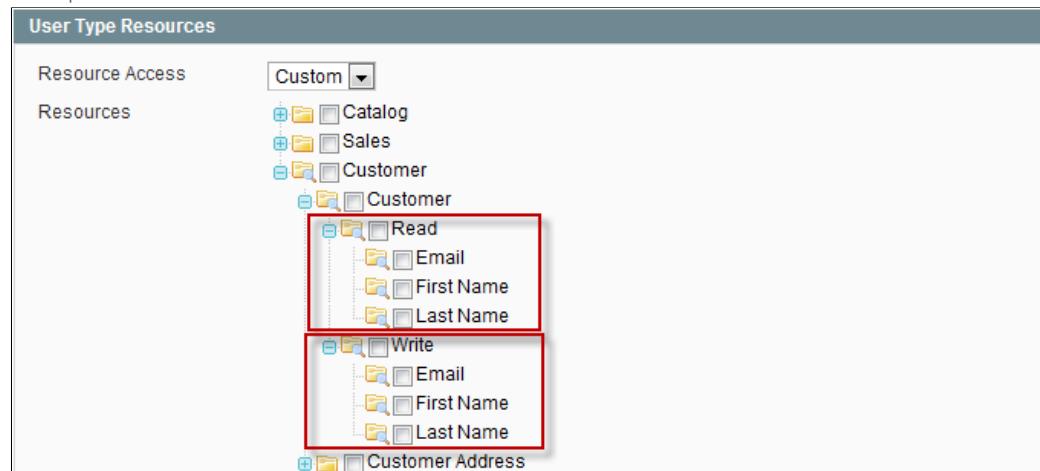
You should remember that ACL is not the same as authentication. ACL is the next step after the authentication is passed successfully. These two concepts are closely connected but the difference lies in the following: authentication is understanding who the user is and ACL is understanding what the user can do.

## ACL Structure

ACL is implemented in a tree structure. There is a tree of resources for each user type. Namely, Admin, Customer, and Guest have their own trees of resources.

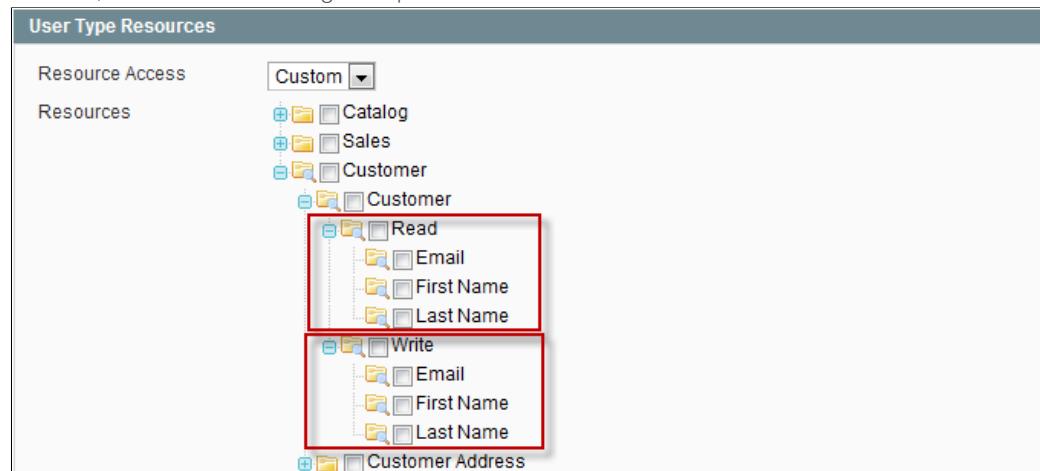
Each ACL entry specifies two instances: a subject and an action the subject can perform.

Example of the resource tree for the Admin role is as follows:



## Read/Write Permissions

All REST resource attributes are divided into two categories: Read and Write. The Read category includes the operation of retrieving. So, when selecting the attributes in the Read category, you specify them for the resource retrieving. The Write category includes the operations of creating and updating. So, when selecting the attributes in the Write category, you specify them for the resource creation and updating. To illustrate the situation, let's take the following example:



## Setting Up ACL

Setting up ACL is performed on two levels:

- Setting up REST roles
- Setting up REST attributes

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Roles Configuration

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▲

### Roles Configuration

#### Attributes Configuration

#### Attributes Description

## Resources ▾

## Response Formats

## Testing REST Resources

## Viewing REST Roles

### Working with Admin Role

#### Adding a New REST Role for Admin

#### Editing an Existing Admin REST Role

#### Deleting an Existing Admin REST Role

#### Assigning a REST Role to Admin

#### Assigning Multiple Users to an Admin REST Role

#### Viewing Users Assigned to an Admin REST Role

#### Unassigning User from the Admin REST Role

### Working with Guest and Customer Roles

#### Editing the Guest REST Role

#### Editing the Customer REST Role

REST roles in Magento are used to limit access to certain resources. Limiting access lies in configuration of a REST role and assigning a user to it. You can select which resources will be available for the user and which will not.

REST roles management consists in the role creation, editing, deleting, and user assignment. Note that REST role creation and deletion is available only for Admin role.

The REST Roles page initially includes two roles: Customer and Guest.

Only API Resources can be edited in the Guest and Customer roles. You cannot change the name or assigned users in these roles. Also, you cannot delete the Guest or Customer role. For example, when editing the Guest Role, the page looks as follows:

## Viewing REST Roles

To view the list of REST roles, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles**.
2. The REST Roles page opens.
3. REST roles are displayed in a grid with the following columns: ID (role ID), Role Name, User Type, and Created At (date and time of the role creation).

## Working with Admin Role

### Adding a New REST Role for Admin

To add a new REST role for Admin, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles**.
2. The REST Roles page opens.
3. In the top right corner, click **Add Admin Role**. The Add New Role page opens.
4. There are two tabs in the Role Information panel on the left: Role Info and Role API Resources.
5. Select the Role Info tab and in the Role Information panel, enter the name for the role to be created in the corresponding **Role Name** field. This field is required.
6. In the Role API Resources tab, in the **Resource Access** drop-down list, select whether the user will have full or custom access by selecting the corresponding **All** or **Custom** options. If you select the **Custom** option, the Resources tree will appear where you will be able to check the required resources and actions.
7. Click **Save Role** in the top right corner to save the role.
8. After you saved the role, a new Role Users tab appears in the Role Information panel on the left. In this tab, you can manage users for the current role. Click **Reset Filter** to view all users to which the role can be assigned.

### Editing an Existing Admin REST Role

To edit an existing Admin REST role, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles**.
2. The REST Roles page opens. In the roles grid, select the Admin role and click it.
3. The Edit Role <role name> page opens. You can edit the following information:
  - **Role Info:** Edit the name of the Admin role by selecting the Role Info tab to the left.
  - **Role API Resources:** Select or clear the resources available for this role.
  - **Role Users:** Assign or remove users for this role.
4. Click **Save Role** in the top right corner to apply changes.

## Deleting an Existing Admin REST Role

To delete an existing Admin REST role, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles**.
2. The REST Roles page opens. In the roles grid, select the required Admin role to be deleted and click it.
3. The Edit Role <role name> page opens. In the top right corner, click **Delete Role**. The role is deleted.

## Assigning a REST Role to Admin

To assign a REST role to admin, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Permissions > Users**.
2. The Users page opens. In the users grid, select the user to which the REST Admin role will be assigned.
3. The Edit User <Name of the User> page opens. In the User Information panel, select the REST Role tab.
4. In the list of REST roles, select the Admin role to be assigned and select the option button next to it.
5. Click **Save User** in the top right corner to save changes.

## Assigning Multiple Users to an Admin REST Role

To assign more than one user to an existing Admin REST role, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles**.
2. The REST Roles page opens. In the REST roles grid, select the Admin role to which users will be assigned.
3. The Edit Role <role name> page opens. In the Role Information panel, select the Role Users tab.
4. In the list of users, click **Reset Filter** to view the list of all users to which the role can be assigned. Select the checkboxes near the users to be assigned to the Admin role.
5. Click **Save Role** in the top right corner to save changes.

## Viewing Users Assigned to an Admin REST Role

To view the list of users assigned to a REST role, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles**.

2. The REST Roles page opens. From the list of roles, select the Admin role whose assigned users you want to view and click it.
3. The Edit Role <role name> page opens. In the Role Information panel on the left, select the Role Users tab.
4. The list of REST role users is displayed in a grid with the following columns: ID, User Name, First Name, and Last Name.

## Unassigning User from the Admin REST Role

To unassign the Admin REST role from a user, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles**.
2. The REST Roles page opens. From the list of roles, select the Admin role from which you want to unassign a user and click it.
3. The Edit Role <role name> page opens. In the Role Information panel on the left, select the Role Users tab.
4. Clear the checkbox next to the user which you want to unassign from the current REST role.
5. Click **Save Role** in the top right corner to apply the changes.

## Working with Guest and Customer Roles

As it has been mentioned before, the Customer and Guest roles cannot be removed and can be only partially edited. You can edit only the resources and actions allowed for the user.

### Editing the Guest REST Role

To edit the Guest REST role, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles**.
2. The REST Roles page opens. From the list of roles, select the Guest role and click it.
3. The Edit Role "Guest" page opens. In the Role Resources panel, edit the required information.
4. Click **Save Role** in the top right corner to apply changes.

### Editing the Customer REST Role

To edit the Customer REST role, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles**.
2. The REST Roles page opens. From the list of roles, select the Customer role and click it.
3. The Edit Role "Customer" page opens. In the Role Resources panel, edit the required information.
4. Click **Save Role** in the top right corner to apply changes.

Create the Magento file system owner

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Attributes Configuration

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

Roles Configuration

[Attributes Configuration](#)

Attributes Description

## Resources ▾

## Response Formats

## Testing REST Resources

[REST Attributes Structure](#)

[Managing Attributes for Guest](#)

[Managing Attributes for Customer](#)

[Managing Attributes for Admin](#)

[Examples](#)

[Limiting Guest Access to Products](#)

[Limiting Customer Access to Products](#)

REST attributes allow specifying additional filters for different types of users. Attributes allow limiting user access more precisely.

## REST Attributes Structure

The REST attributes tree includes the following elements (as subnodes):

- Name of the resource
  - Read permissions (includes all elements available for the current resource)
  - Write permissions (includes all elements available for the current resource)

The Resources tree may be too immense. To avoid scrolling down when searching for the required resource, you can fold the nodes for better representation.

## Managing Attributes for Guest

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Attributes**.
2. The REST Attributes page opens. From the list of user types, select the **Guest** type and click it.
3. The page for editing attribute rules opens.
4. In the User Type Resources panel, in the **Resource Access** drop-down list, select whether all or some specific resources will be limited to a **Guest** type of user by selecting the corresponding **All** or **Custom** options.
5. If you select the **Custom** option, the resources tree appears.
6. Select the required options and click **Save** in the top right corner to apply changes.

## Managing Attributes for Customer

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Attributes**.
2. The REST Attributes page opens. From the list of user types, select the **Customer** type and click it.
3. The page for editing attribute rules opens.
4. In the User Type Resources panel, in the **Resource Access** drop-down list, select whether all or some specific resources will be limited to a **Customer** type of user by selecting the corresponding **All** or

**Custom** options.

5. If you select the **Custom** option, the resources tree appears. Some resources have options for selecting read and write permissions.
6. Select the required options and click **Save** in the top right corner to apply changes.

## Managing Attributes for Admin

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Attributes**.
2. The REST Attributes page opens. From the list of user types, select the **Admin** type and click it.
3. The page for editing attribute rules opens.
4. In the User Type Resources panel, in the **Resource Access** drop-down list, select whether all or some specific resources will be limited to an **Admin** type of user by selecting the corresponding **All** or **Custom** options.
5. If you select the **Custom** option, the resources tree appears. Each resource has options for selecting read and write permissions.
6. Select the required options and click **Save** in the top right corner to apply changes.

## Examples

This section provides some examples of limiting Guest and Customer access to certain resource elements.

### Limiting Guest Access to Products

To allow Guests (users that are not registered in the Magento system) view only product name and final price with tax, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles** and select the Guest role.
2. In the **Role API Resources**, specify the Retrieve option for the Product resource.
3. Click **Save Role** on the top right corner to save the role.
4. On the Magento Admin Panel menu, select **System > Web Services > REST - Attributes** and select **Guest** in the list of User Types.
5. In the Resources tree, navigate to the **Catalog Product** node. In the Read subnode, select the **Name** and **Final Price With Tax** options.
6. Click **Save** in the top right corner to save the selected attributes.

### Limiting Customer Access to Products

To allow Customers (users that are registered in the Magento system) view only product name and final price with tax, perform the following steps:

1. On the Magento Admin Panel menu, select **System > Web Services > REST - Roles** and select the Customer role.
2. In the **Role API Resources**, specify the Retrieve option for the Product resource.
3. Click **Save Role** on the top right corner to save the role.

4. On the Magento Admin Panel menu, select **System > Web Services > REST - Attributes** and select **Customer** in the list of User Types.
5. In the Resources tree, navigate to the **Catalog Product** node. In the Read subnode, select the **Name** and **Final Price With Tax** options.
6. Click **Save** in the top right corner to save the selected attributes.

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attributes Description

[Edit this page on GitHub](#) 

### Introduction

### Authentication

### Common HTTP Status Codes

### HTTP Methods

### GET Filters

### Permission Settings

Roles Configuration

Attributes Configuration

[Attributes Description](#)

### Resources

### Response Formats

### Testing REST Resources

### Order/Orders

[Order Addresses](#)

[Order Items](#)

[Stock Item](#)

[Customer](#)

[Customer Address](#)

[Product](#)

[Product Category](#)

[Product Image](#)

The following table describes REST attributes that can be managed in the Magento Admin Panel. To access these attributes, go to **System > Web Services > REST Attributes** and select the type of the user for which attributes will be managed.

## Order/Orders

| Attribute Name  | Attribute Description                                                                                                                 | Notes |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------|-------|
| Order ID        | Sales order ID                                                                                                                        |       |
| Order Date      | Date when the sales order was placed                                                                                                  |       |
| Order Status    | Sales order status. Can have the following values: Pending, Processing, Complete, Closed, Holded, Pending PayPal, and Payment Review. |       |
| Shipping Method | Shipping method selected during the checkout process (e.g., Flat rate - Fixed)                                                        |       |
| Payment Method  | Payment method selected during the checkout process (e.g., Check/money order)                                                         |       |
| Base Currency   | Base currency code (e.g., USD)                                                                                                        |       |

|                                      |                                                                                             |                                                |
|--------------------------------------|---------------------------------------------------------------------------------------------|------------------------------------------------|
| Order Currency                       | Order currency code (e.g., EUR)                                                             |                                                |
| Store Name                           | Name of the store from which the order was placed                                           |                                                |
| Placed from IP                       | IP address from which the order was placed                                                  |                                                |
| Store Currency to Base Currency Rate | Store currency to base currency rate                                                        |                                                |
| Subtotal                             | Subtotal amount in order currency (excluding shipping and tax)                              |                                                |
| Subtotal Including Tax               | Subtotal amount including tax (in order currency)                                           |                                                |
| Discount                             | Discount amount applied in the sales order in order currency                                |                                                |
| Grand Total to Be Charged            | Total amount of money to be paid for the order in base currency (including tax)             |                                                |
| Grand Total                          | Grand total amount in order currency (including tax and shipping)                           |                                                |
| Shipping                             | Shipping amount applied in the sales order in order currency                                |                                                |
| Shipping Including Tax               | Shipping amount including tax (in order currency)                                           |                                                |
| Shipping Tax                         | Tax amount for shipping in order currency                                                   |                                                |
| Tax Amount                           | Tax amount applied in the sales order in order currency                                     |                                                |
| Tax Name                             | Name of the applied tax                                                                     |                                                |
| Tax Rate                             | Tax rate applied in the order (in order currency)                                           |                                                |
| Gift Cards Amount                    | Gift card pricing amount                                                                    | This attribute is available only in Magento EE |
| Reward Points Balance                | Reward points amount (that can be converted to currency)                                    | This attribute is available only in Magento EE |
| Reward Currency Amount               | Reward currency amount                                                                      | This attribute is available only in Magento EE |
| Coupon Code                          | Coupon code that was applied in the order                                                   |                                                |
| Base Discount                        | Amount of applied discount in base currency                                                 |                                                |
| Base Subtotal                        | Subtotal amount for all products in the order in base currency (excluding tax and shipping) |                                                |
| Base Shipping                        | Amount of money to be paid for shipping in base currency                                    |                                                |
| Base Shipping Tax                    | Tax amount for shipping in base currency                                                    |                                                |
| Base Tax Amount                      | Tax amount applied to the order items in base currency                                      |                                                |
| Total Paid                           | Total amount paid for the order (in order currency)                                         |                                                |
| Base Total Paid                      | Total amount paid for the order (in base currency)                                          |                                                |
| Total Refunded                       | Total refunded amount in order currency                                                     |                                                |

|                             |                                                                                                          |                                                |
|-----------------------------|----------------------------------------------------------------------------------------------------------|------------------------------------------------|
| Base Total Refunded         | Total amount refunded for the order (in base currency)                                                   |                                                |
| Base Subtotal Including Tax | Subtotal amount including tax but excluding the discount amount (in base currency)                       |                                                |
| Base Total Due              | The rest of the money to be paid for the order in base currency (e.g., when partial invoice is applied)  |                                                |
| Total Due                   | The rest of the money to be paid for the order in order currency (e.g., when partial invoice is applied) |                                                |
| Shipping Discount           | Discount amount for shipping (in order currency)                                                         |                                                |
| Base Shipping Discount      | Discount amount for shipping (in base currency)                                                          |                                                |
| Discount Description        | Discount code (coupon code applied in the order)                                                         |                                                |
| Customer Balance            | Customer balance (in order currency)                                                                     |                                                |
| Base Customer Balance       | Customer balance (in base currency)                                                                      |                                                |
| Base Gift Cards Amount      | Gift card pricing amount (in base currency)                                                              | This attribute is available only in Magento EE |
| Base Rewards Currency       | Reward currency amount (in base currency)                                                                | This attribute is available only in Magento EE |

## Order Addresses

| Attribute Name       | Attribute Description           |
|----------------------|---------------------------------|
| Customer Last Name   | Customer last name              |
| Customer First Name  | Customer first name             |
| Customer Middle Name | Customer middle name or initial |
| Customer Prefix      | Customer prefix                 |
| Customer Suffix      | Customer suffix                 |
| Company              | Company name                    |
| Street               | Street address                  |
| City                 | City                            |
| State                | State                           |
| ZIP/Postal Code      | ZIP or postal code              |
| Country              | Country name                    |
| Phone Number         | Customer phone number           |

## Address Type

Address type. Can have the following values: billing or shipping

## Order Items

| Attribute Name                   | Attribute Description                                                  |
|----------------------------------|------------------------------------------------------------------------|
| Base Discount Amount             | Discount amount applied to the row in base currency                    |
| Base Item Subtotal               | Row subtotal in base currency                                          |
| Base Item Subtotal Including tax | Row subtotal including tax in base currency                            |
| Base Original Price              | Original item price in base currency                                   |
| Base Price                       | Item price in base currency                                            |
| Base Price Including tax         | Item price including tax in base currency                              |
| Base Tax Amount                  | Tax amount applied to the row in base currency                         |
| Canceled Qty                     | Number of canceled order items                                         |
| Discount Amount                  | Discount amount applied to the row in order currency                   |
| Invoiced Qty                     | Number of invoiced order items                                         |
| Item Subtotal                    | Row subtotal in order currency                                         |
| Item Subtotal Including Tax      | Row subtotal including tax in order currency                           |
| Order Item ID                    | Order item ID                                                          |
| Ordered Qty                      | Number of ordered items                                                |
| Original Price                   | Original item price in order currency                                  |
| Parent Order Item ID             | ID of the configurable product to which the simple product is assigned |
| Price                            | Item price in order currency                                           |
| Price Including Tax              | Item price including tax in order currency                             |
| Product and Custom Options Name  | Name of the product (custom options name)                              |
| Refunded Qty                     | Number of refunded order items                                         |
| SKU                              | Product SKU                                                            |
| Shipped Qty                      | Number of shipped order items                                          |
| Tax Amount                       | Tax amount applied to the row in order currency                        |
| Tax Percent                      | Tax percent applied to the row                                         |

## Stock Item

| Attribute Name                                 | Attribute Description                                                                                   |
|------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| Automatically Return Credit Memo Item to Stock | Defines whether products can be automatically returned to stock when the refund for an order is created |
| Backorders                                     | Defines whether the customer can place the order for products that are out of stock at the              |

|                                                              |                                                                                                                                                                         |
|--------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                              | moment. Can have the following values: 0 - No Backorders, 1 - Allow Qty Below 0, and 2 - Allow Qty Below 0 and Notify Customer                                          |
| Can Be Divided into Multiple Boxes for Shipping              | Defines whether the stock items can be divided into multiple boxes for shipping                                                                                         |
| Enable Qty Increments                                        | Defines whether the customer can add products only in increments to the shopping cart                                                                                   |
| Item ID                                                      | Stock item ID                                                                                                                                                           |
| Low Stock Date                                               | Date when the number of stock items became lower than the number defined in the Notify for Quantity Below option                                                        |
| Manage Stock                                                 | Choose whether to view and specify the product quantity and availability and whether the product is in stock management. Can have the following values: 0 - No, 1 - Yes |
| Maximum Qty Allowed in Shopping Cart                         | Maximum number of items in the shopping cart to be sold                                                                                                                 |
| Minimum Qty Allowed in Shopping Cart                         | Minimum number of items in the shopping cart to be sold                                                                                                                 |
| Notify for Quantity Below                                    | The number of inventory items below which the customer will be notified via the RSS feed                                                                                |
| Product ID                                                   | Product ID                                                                                                                                                              |
| Qty                                                          | Quantity of stock items for the current product                                                                                                                         |
| Qty Increments                                               | The product quantity increment value                                                                                                                                    |
| Qty Uses Decimals                                            | Choose whether the product can be sold using decimals (e.g., you can buy 2.5 product)                                                                                   |
| Qty for Item's Status to Become Out of Stock                 | Quantity for stock items to become out of stock                                                                                                                         |
| Stock Availability                                           | Defines whether the product is available for selling. Can have the following values: 0 - Out of Stock, 1 - In Stock                                                     |
| Stock ID                                                     | Stock ID                                                                                                                                                                |
| Use Config Settings for Backorders                           | Choose whether the Config settings will be applied for the Backorders option                                                                                            |
| Use Config Settings for Enable Qty Increments                | Choose whether the Config settings will be applied for the Enable Qty Increments option                                                                                 |
| Use Config Settings for Manage Stock                         | Choose whether the Config settings will be applied for the Manage Stock option                                                                                          |
| Use Config Settings for Maximum Qty Allowed in Shopping Cart | Choose whether the Config settings will be applied for the Maximum Qty Allowed in Shopping Cart option                                                                  |
| Use Config Settings for Minimum Qty Allowed in Shopping Cart | Choose whether the Config settings will be applied for the Minimum Qty Allowed in Shopping Cart option                                                                  |
| Use Config Settings for Notify for Quantity Below            | Choose whether the Config settings will be applied for the Notify for Quantity Below option                                                                             |
| Use Config Settings for Qty Increments                       | Choose whether the Config settings will be applied for the Qty Increments option                                                                                        |

|                                                                      |                                                                                                                |
|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| Use Config Settings for Qty for Item's Status to Become Out of Stock | Choose whether the Config settings will be applied for the Qty for Item's Status to Become Out of Stock option |
|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|

**Notes:** The Admin user type has restrictions concerning the WRITE operations for definite stock item attributes. These are as follows:

|                |       |
|----------------|-------|
| Attribute Name | Admin |
| Item ID        | No    |
| Product ID     | No    |
| Stock ID       | No    |
| Low Stock Date | No    |

However, these attributes are available for READ operations.

## Customer

| Attribute Name                 | Attribute Description                                                      |
|--------------------------------|----------------------------------------------------------------------------|
| Customer ID                    | Customer ID                                                                |
| Last Logged In                 | Date when the customer was logged in last                                  |
| Is Confirmed                   | Defines whether the email confirmation is sent to the customer             |
| Created At                     | Date when the customer was created                                         |
| Associate to Website           | Website ID to which the customer is associated                             |
| Created From                   | Store view from which the customer was created                             |
| Group                          | Customer group ID                                                          |
| Disable automatic group change | Defines whether the automatic group change will be applied to the customer |
| Prefix                         | Customer prefix                                                            |
| First Name                     | Customer first name                                                        |
| Middle Name/Initial            | Customer middle name or initial                                            |
| Last Name                      | Customer last name                                                         |
| Suffix                         | Customer suffix                                                            |
| Email                          | Customer email address                                                     |
| Date Of Birth                  | Customer date of birth                                                     |
| Tax/VAT Number                 | Customer tax or VAT number                                                 |
| Gender                         | Customer gender (male or female)                                           |

## Customer Address

| Attribute Name | Attribute Description |
|----------------|-----------------------|
| City           | City name             |

|                             |                                                           |
|-----------------------------|-----------------------------------------------------------|
| Company                     | Company name                                              |
| Country                     | Country                                                   |
| Customer Address ID         | Customer address ID                                       |
| Fax                         | Fax number                                                |
| First Name                  | Customer first name                                       |
| Is Default Billing Address  | Defines whether the address is a default one for billing  |
| Is Default Shipping Address | Defines whether the address is a default one for shipping |
| Last Name                   | Customer last name                                        |
| Middle Name/Initial         | Customer middle name or initial                           |
| Prefix                      | Customer prefix                                           |
| State/Province              | Customer state/region                                     |
| Street Address              | Customer street address                                   |
| Suffix                      | Customer suffix                                           |
| Telephone                   | Customer phone number                                     |
| VAT Number                  | Customer VAT number                                       |
| ZIP/Postal Code             | Customer ZIP or postal code                               |

## Product

Attributes for the product resource are divided into those available for the Admin type of user and those available for the Customer and Guest types of user.

| Attribute Name     | Attribute Description                                                                                                      | Notes                    |
|--------------------|----------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Product ID         | Product ID                                                                                                                 | Available only for Admin |
| name               | Product Name                                                                                                               |                          |
| Product Type       | Product type. Can have the following values: Simple, Grouped, Configurable, Virtual, Bundle, or Downloadable               |                          |
| Attribute Set Name | Name of the attribute set which the product is based on                                                                    | Available only for Admin |
| sku                | Product SKU                                                                                                                |                          |
| price              | Product price                                                                                                              |                          |
| visibility         | Product visibility in the store. Can have the following values: Catalog, Search; Search; Catalog; Not Visible Individually | Available only for Admin |
| description        | Product description                                                                                                        |                          |
| short_description  | Product short description                                                                                                  |                          |
| weight             | Product weight                                                                                                             | Available only for Admin |
| news_from_date     | Date starting from which the product is promoted as a new                                                                  | Available only for Admin |

|                                       |                                                                                                                                              |                                                             |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
|                                       | product                                                                                                                                      |                                                             |
| news_to_date                          | Date till which the product is promoted as a new product                                                                                     | Available only for Admin                                    |
| status                                | Product status in the store. Can have the following values: Enabled or Disabled                                                              | Available only for Admin                                    |
| url_key                               | A friendly URL path for the product                                                                                                          | Available only for Admin                                    |
| Create Permanent Redirect for Old URL | Defines whether the redirect to an original URL will be applied (when the existing URL for a product is edited)                              | Available only for Admin; available only for product update |
| country_of_manufacture                | Product country of manufacture                                                                                                               | Available only for Admin                                    |
| is_returnable                         | Defines whether the product can be returned                                                                                                  | Available only for Admin                                    |
| special_price                         | Product special price                                                                                                                        | Available only for Admin                                    |
| special_from_date                     | Date starting from which the special price will be applied for the product                                                                   | Available only for Admin                                    |
| special_to_date                       | Date till which the special price will be applied for the product                                                                            | Available only for Admin                                    |
| group_price                           | Product group price                                                                                                                          | Available only for Admin                                    |
| tier_price                            | Product tier price                                                                                                                           |                                                             |
| msrp_enabled                          | The Apply MAP option. Defines whether the price in the catalog in the frontend is substituted with a Click for price link                    | Available only for Admin                                    |
| msrp_display_actual_price_type        | Defines how the price will be displayed in the frontend. Can have the following values: In Cart, Before Order Confirmation, and On Gesture   | Available only for Admin                                    |
| msrp                                  | The Manufacturer's Suggested Retail Price option. The price that a manufacturer suggests to sell the product at                              | Available only for Admin                                    |
| enable_googlecheckout                 | Defines whether the product can be purchased with the help of the Google Checkout payment service. Can have the following values: Yes and No | Available only for Admin                                    |
| tax_class_id                          | The product tax class to which the product will be associated                                                                                | Available only for Admin                                    |
| meta_title                            | Product meta title                                                                                                                           |                                                             |
| meta_keyword                          | Product meta keywords                                                                                                                        |                                                             |
| meta_description                      | Product meta description                                                                                                                     |                                                             |
| custom_design                         | Custom design applied for the product page                                                                                                   | Available only for Admin                                    |
| custom_design_from                    | Date starting from which the custom design will be applied for the product page                                                              | Available only for Admin                                    |
| custom_design_to                      | Date till which the custom design will be applied for the product page                                                                       | Available only for Admin                                    |
| custom_layout_update                  | An XML block to alter the page layout                                                                                                        | Available only for Admin                                    |

|                                             |                                                                                                                                                 |                                                                          |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| page_layout                                 | Page template that can be applied to the product page                                                                                           | Available only for Admin                                                 |
| options_container                           | Defines how the custom options for the product will be displayed. Can have the following values: Block after Info Column or Product Info Column | Available only for Admin                                                 |
| gift_message_available                      | Defines whether the gift message is available for the product                                                                                   | Available only for Admin                                                 |
| Use Config Settings for Allow Gift Message  | Defines whether the configuration settings will be used for the Allow Gift Message option                                                       | Available only for Admin                                                 |
| gift_wrapping_available                     | Defines whether the gift wrapping is available for the product                                                                                  | Available only for Admin. This attribute is available in Magento EE      |
| Use Config Settings for Allow Gift Wrapping | Defines whether the configuration settings will be used for the Allow Gift Wrapping option                                                      | Available only for Admin. This attribute is available in Magento EE      |
| gift_wrapping_price                         | Price for the gift wrapping (available in Magento EE)                                                                                           | Available only for Admin                                                 |
| Inventory Data                              | Product inventory data                                                                                                                          | Available only for Admin                                                 |
| Custom attr                                 | Product custom attributes                                                                                                                       | The customer can see only attributes that are set as visible on frontend |
| Regular Price                               | The original product price displayed in the frontend                                                                                            | Available only for Customer and Guest                                    |
| Final Price                                 | The final product price                                                                                                                         | Available only for Customer and Guest                                    |
| Final Price with Tax                        | The final product price with tax                                                                                                                | Available only for Customer and Guest                                    |
| Final Price Without Tax                     | The final product price without tax                                                                                                             | Available only for Customer and Guest                                    |
| Stock Status                                | The product stock status (availability)                                                                                                         | Available only for Customer and Guest                                    |
| Product Is Saleable                         | Defines whether the product can be sold                                                                                                         | Available only for Customer and Guest                                    |
| Total Reviews Number                        | The number of all reviews for a product                                                                                                         | Available only for Customer and Guest                                    |
| Product URL Link                            | A link to the product without the assigned category                                                                                             | Available only for Customer and Guest                                    |
| Buy Now Link                                | A link that adds a product to the shopping cart                                                                                                 | Available only for Customer and Guest                                    |
| Product Has Custom Options                  | Defines whether the product has custom options or not                                                                                           | Available only for Customer and Guest                                    |
| Default Product Image                       | Default product image                                                                                                                           | Available only for Customer and Guest                                    |

## Product Category

| Attribute Name | Attribute Description                               |
|----------------|-----------------------------------------------------|
| Category ID    | ID of the category to which the product is assigned |

## Product Image

| Attribute Name | Attribute Description                                                                         | Notes                               |
|----------------|-----------------------------------------------------------------------------------------------|-------------------------------------|
| Exclude        | Defines whether the image will associate only to one of the three image types.                |                                     |
| ID             | Image file ID                                                                                 | Available only for READ operations  |
| Label          | A label that will be displayed on the frontend when pointing to the image                     |                                     |
| Position       | The Sort Order option. The order in which the images are displayed in the MORE VIEWS section. |                                     |
| Type           | Image type. Can have the following values: Base Image, Small Image, or Thumbnail.             |                                     |
| URL            | Image file URL path                                                                           | Available only for READ operations  |
| File Content   | Image file content (base_64 encoded)                                                          | Available only for WRITE operations |
| File MIME Type | File MIME type. Can have the following values: image/jpeg, image/png, image/gif, etc.         | Available only for WRITE operations |
| File Name      | Image file name                                                                               | Available only for WRITE operations |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Resources

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ^

Inventory

## Orders ▾

## Products ▾

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

## Products

Product Categories

Product Images

Product Websites

Customers

Customer Addresses

Inventory

Sales Orders

Order Addresses

Order Comments

Order Items

The Magento REST API allows you to manage customers, customer addresses, sales orders, inventory, and products. REST API is organized into the following categories:

## Products

Retrieve the list of products, create, update, delete a product.

**Resource Structure:** `http://magentohost/api/rest/products`

## Product Categories

Retrieve the list of categories assigned to a product, assign and unassign the category from a product.

**Resource Structure:** `http://magentohost/api/rest/products/:id/categories`

## Product Images

Retrieve the list of images assigned to a product, add, update, remove an image to/from a product.

**Resource Structure:** `http://magentohost/api/rest/products/:id/images`

## **Product Websites**

Retrieve the list of websites assigned to a product, assign, unassign a website to/from a product.

**Resource Structure:** <http://magentohost/api/rest/products/:id/websites>

## **Customers**

Retrieve the list of customers, create, delete a customer, and update the customer information.

**Resource Structure:** <http://magentohost/api/rest/customers>

## **Customer Addresses**

Retrieve the list of customer addresses, create, update, and delete the customer address.

**Resource Structure:** <http://magentohost/api/rest/customers/:id/addresses>

## **Inventory**

Retrieve the list of stock items, update required stock items.

**Resource Structure:** <http://magentohost/api/rest/stockitems>

## **Sales Orders**

Retrieve the list of sales orders with detailed information on order addresses, items, and comments.

**Resource Structure:** <http://magentohost/api/rest/orders>

## **Order Addresses**

Retrieve information on order billing and shipping addresses.

**Resource Structure:** <http://magentohost/api/rest/orders/:id/addresses>

## **Order Comments**

Retrieve information on the specified order comments.

**Resource Structure:** <http://magentohost/api/rest/orders/:id/comments>

## **Order Items**

Retrieve information on specified order items.

**Resource Structure:** <http://magentohost/api/rest/orders/:id/items>

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Inventory

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ▾

### Inventory

### Orders ▾

### Products ▾

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

### REST API: Stock Items

URI: /stockitems

HTTP Method: GET /stockitems

HTTP Method: PUT /stockitems

### REST API: Stock Item

URI: /stockitems/:id

HTTP Method : GET /stockitems/:id

HTTP Method : PUT /stockitems/:id

HTTP Method : DELETE /stockitems/:id

**!** **Important:** The default response for all APIs is JSON; however, this page currently uses mostly XML examples. We don't maintain this documentation much anymore so if you'd like to contribute some JSON, we'll acknowledge your help on this page. [Edit this page on GitHub](#) if you can help us out.

## REST API: Stock Items

**URI:** /stockitems

Allows you to manage existing stock items. Inventory management is available only for Admin.

**URL Structure:** <http://magentohost/api/rest/stockitems>

**Version:** 1

### HTTP Method: GET /stockitems

**Description:** Allows you to retrieve the list of existing stock items.

**Notes:** The list of attributes that will be returned for stock items is configured in the Magento Admin Panel.

**Authentication:** Admin

**Default Format:** JSON

**Parameters :**

*No Parameters*

#### Response Example: XML

```
GET http://magentohost/api/rest/stockitems
```

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <item_id>1</item_id>
 <qty>100.0000</qty>
 <backorders>0</backorders>
 <min_sale_qty>1.0000</min_sale_qty>
 <max_sale_qty>0.0000</max_sale_qty>
 <low_stock_date></low_stock_date>
 <manage_stock>0</manage_stock>
 <stock_status_changed_auto>0</stock_status_changed_auto>
 <enable_qty_increments>0</enable_qty_increments>
 </data_item>
 <data_item>
 <item_id>2</item_id>
 <qty>100.0000</qty>
 <backorders>0</backorders>
 <min_sale_qty>1.0000</min_sale_qty>
 <max_sale_qty>0.0000</max_sale_qty>
 <low_stock_date></low_stock_date>
 <manage_stock>0</manage_stock>
 <stock_status_changed_auto>0</stock_status_changed_auto>
 <enable_qty_increments>0</enable_qty_increments>
 </data_item>
 <data_item>
 <item_id>3</item_id>
 <qty>1.0000</qty>
 <backorders>0</backorders>
 <min_sale_qty>1.0000</min_sale_qty>
 <max_sale_qty>0.0000</max_sale_qty>
 <low_stock_date></low_stock_date>
 <manage_stock>0</manage_stock>
 <stock_status_changed_auto>0</stock_status_changed_auto>
 <enable_qty_increments>0</enable_qty_increments>
 </data_item>
 <data_item>
 <item_id>4</item_id>
 <qty>0.0000</qty>
 <backorders>0</backorders>
 <min_sale_qty>1.0000</min_sale_qty>
 <max_sale_qty>0.0000</max_sale_qty>
 <low_stock_date></low_stock_date>
 <manage_stock>0</manage_stock>
 <stock_status_changed_auto>1</stock_status_changed_auto>
 <enable_qty_increments>0</enable_qty_increments>
 </data_item>
</magento_api>
```

#### Response Example: JSON

```
GET http://magentohost/api/rest/stockitems
```

```
[{"item_id": "1", "qty": "100.0000", "backorders": "0", "min_sale_qty": "1.0000", "max_sale_qty": "0"}, {"item_id": "2", "qty": "100.0000", "backorders": "0", "min_sale_qty": "1.0000", "max_sale_qty": "0"}]
```

```
{"item_id": "3", "qty": "1.0000", "backorders": "0", "min_sale_qty": "1.0000", "max_sale_qty": "0"
{"item_id": "4", "qty": "0.0000", "backorders": "0", "min_sale_qty": "1.0000", "max_sale_qty": "0"}
```

## HTTP Method: PUT /stockitems

**Description:** Allows you to update existing stock items.

**Authentication:** Admin

**Default Format:** JSON

**Notes:** The Content-Type: text/xml parameter must be added to the request header.

### Parameters:

| Name                        | Description                                                                                                                                                                 | Type   | Example Value       |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|---------------------|
| item_id                     | Item ID                                                                                                                                                                     | int    | 1                   |
| product_id                  | Product ID                                                                                                                                                                  | int    | 1                   |
| stock_id                    | Stock ID                                                                                                                                                                    | int    | 1                   |
| qty                         | Quantity of stock items for the current product                                                                                                                             | string | 20                  |
| min_qty                     | Quantity for stock items to become out of stock                                                                                                                             | string | 0                   |
| use_config_min_qty          | Choose whether the Config settings will be applied for the Qty for Item's Status to Become Out of Stock option                                                              | int    | 1                   |
| is_qty_decimal              | Choose whether the product can be sold using decimals (e.g., you can buy 2.5 product)                                                                                       | int    | 0                   |
| backorders                  | The customer can place the order for products that are out of stock at the moment (0 - No Backorders, 1 - Allow Qty Below 0, and 2 - Allow Qty Below 0 and Notify Customer) | int    | 0                   |
| use_config_backorders       | Choose whether the Config settings will be applied for the Backorders option                                                                                                | int    | 1                   |
| min_sale_qty                | Minimum number of items in the shopping cart to be sold                                                                                                                     | string | 10                  |
| use_config_min_sale_qty     | Choose whether the Config settings will be applied for the Minimum Qty Allowed in Shopping Cart option                                                                      | int    | 0                   |
| max_sale_qty                | Maximum number of items in the shopping cart to be sold                                                                                                                     | string | 100                 |
| use_config_max_sale_qty     | Choose whether the Config settings will be applied for the Maximum Qty Allowed in Shopping Cart option                                                                      | int    | 0                   |
| is_in_stock                 | Defines whether the product is available for selling (0 - Out of Stock, 1 - In Stock)                                                                                       | int    | 1                   |
| low_stock_date              | Date when the number of stock items became lower than the number defined in the Notify for Quantity Below option                                                            | string | 2012-02-24 12:37:51 |
| notify_stock_qty            | The number of inventory items below which the customer will be notified via the RSS feed                                                                                    | string | 10                  |
| use_config_notify_stock_qty | Choose whether the Config settings will be applied for the Notify for Quantity Below option                                                                                 | int    | 0                   |

|                           |                                                                                                                                           |        |   |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|--------|---|
| manage_stock              | Choose whether to view and specify the product quantity and availability and whether the product is in stock management( 0 - No, 1 - Yes) | int    | 0 |
| use_config_manage_stock   | Choose whether the Config settings will be applied for the Manage Stock option                                                            | int    | 1 |
| stock_status_changed_auto | Defines whether products can be automatically returned to stock when the refund for an order is created                                   | int    | 0 |
| use_config_qty_increments | Choose whether the Config settings will be applied for the Enable Qty Increments option                                                   | int    | 1 |
| qty_increments            | The product quantity increment value                                                                                                      | string | 5 |
| use_config_enable_qty_inc | Choose whether the Config settings will be applied for the Qty Increments option                                                          | int    | 1 |
| enable_qty_increments     | Defines whether the customer can add products only in increments to the shopping cart                                                     | int    | 0 |
| is_decimal_divided        | Defines whether the stock items can be divided into multiple boxes for shipping.                                                          | int    | 0 |

#### Example: XML

```
PUT http://magentohost/api/rest/stockitems
```

#### Request Body:

```
<?xml version="1.0"?>
<magento_api>
 <data_item item_id="157">
 <product_id>262</product_id>
 <stock_id>1</stock_id>
 <qty>100.0000</qty>
 <min_qty>0.0000</min_qty>
 </data_item>
 <data_item item_id="158">
 <product_id>263</product_id>
 <stock_id>1</stock_id>
 <qty>100.0000</qty>
 <min_qty>0.0000</min_qty>
 </data_item>
 <data_item item_id="159">
 <product_id>264</product_id>
 <stock_id>1</stock_id>
 <qty>120.0000</qty>
 <min_qty>0.0000</min_qty>
 </data_item>
 <data_item item_id="153">
 <product_id> </product_id>
 <qty>110.0000</qty>
 <min_qty>0.0000</min_qty>
 </data_item>
</magento_api>
```

#### Response Body:

```
<?xml version="1.0"?>
<magento_api>
 <success>
```

```

<data_item>
 <message>Resource updated successful.</message>
 <code>200</code>
 <item_id>157</item_id>
</data_item>
</success>
<error>
 <data_item>
 <message>Resource not found.</message>
 <code>404</code>
 <item_id>158</item_id>
 </data_item>
 <data_item>
 <message>Resource not found.</message>
 <code>404</code>
 <item_id>159</item_id>
 </data_item>
 <data_item>
 <message>Empty value for "product_id" in request.</message>
 <code>400</code>
 <item_id>153</item_id>
 </data_item>
</error>
</magento_api>

```

## REST API: Stock Item

URI: /stockitems/:id

Allows you to update, delete, or retrieve information on a single stock item.

**Notes:** The list of attributes that will be returned for stock items is configured in the Magento Admin Panel.

**URL Structure:** <http://magentohost/api/rest/stockitems/:id>

**Version:** 1

### HTTP Method : GET /stockitems/:id

**Description:** Allows you to retrieve the stock item information.

**Authentication:** Admin

**Default Format:** JSON

### Response Example: XML

```
GET http://magentohost/api/rest/stockitems/1
```

#### Response Body:

```

<?xml version="1.0"?>
<magento_api>
 <item_id>1</item_id>
 <product_id>1</product_id>
 <stock_id>1</stock_id>
 <qty>200.0000</qty>
 <min_qty>0.0000</min_qty>
 <use_config_min_qty>1</use_config_min_qty>
 <is_qty_decimal>1</is_qty_decimal>
 <backorders>0</backorders>
 <use_config_backorders>1</use_config_backorders>
 <min_sale_qty>1.0000</min_sale_qty>
 <use_config_min_sale_qty>1</use_config_min_sale_qty>
 <max_sale_qty>0.0000</max_sale_qty>

```

```

<use_config_max_sale_qty>1</use_config_max_sale_qty>
<is_in_stock>1</is_in_stock>
<low_stock_date></low_stock_date>
<notify_stock_qty>10.0000</notify_stock_qty>
<use_config_notify_stock_qty>0</use_config_notify_stock_qty>
<manage_stock>0</manage_stock>
<use_config_manage_stock>1</use_config_manage_stock>
<stock_status_changed_auto>0</stock_status_changed_auto>
<use_config_qty_increments>1</use_config_qty_increments>
<qty_increments>0.0000</qty_increments>
<use_config_enable_qty_inc>1</use_config_enable_qty_inc>
<enable_qty_increments>0</enable_qty_increments>
<is_decimal_divided>1</is_decimal_divided>
</magento_api>

```

### Response Example: JSON

GET http://magentohost/api/rest/stockitems/157

#### Response Body:

```
{"item_id":"1","product_id":"1","stock_id":"1","qty":"200.0000","min_qty":"0.0000","use_
```

### HTTP Method : PUT /stockitems/:id

**Description:** Allows you to update existing stock item data.

**Notes:** The Content-Type: text/xml parameter must be added to the request header.

**Authentication:** Admin

**Default Format:** JSON

**Parameters:**

*Enter only those parameters which you want to update.*

#### Example: XML

PUT http://magentohost/api/rest/stockitems/1

#### Request Body:

```
<?xml version="1.0"?>
<magento_api>
 <qty>99</qty>
</magento_api>
```

#### Example: JSON

PUT http://magentohost/api/rest/stockitems/1

#### Request Body:

```
{
 "qty": "99"
}
```

### HTTP Method : DELETE /stockitems/:id

**Description:** Not allowed. The DELETE method is not allowed because you cannot delete a stock item. The required stock item is deleted together with the product which it is associated to.

#### Possible HTTP Status Codes:

| Error Code | Error Message                                       | Error Description                                                     |
|------------|-----------------------------------------------------|-----------------------------------------------------------------------|
| 200        | Resource updated successful.                        | The required resource was successfully updated.                       |
| 404        | Resource not found.                                 | The required resource is not found or does not exist.                 |
| 400        | Empty value for <name of the parameter> in request. | Value is not defined for the specified parameter in the request body. |
| 400        | Invalid value for "item_id" in request.             | The specified value for "item_id" is not valid.                       |
| 400        | Missing <name of the parameter> in request.         | The specified parameter is missing in the request body.               |
| 500        | Resource internal error.                            | Resource internal error.                                              |

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Sales Orders

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ^

Inventory

### Orders ^

Order Addresses

Order Comments

Order Items

### Products ▾

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

### REST API: Sales Orders

URI: /orders

HTTP Method: GET /orders

HTTP Method: POST /orders

HTTP Method: PUT /orders

HTTP Method: DELETE /orders

### REST API: Sales Order

URI: /orders/:id

HTTP Method: GET /orders/:id

HTTP Method: POST /orders/:id

HTTP Method: PUT /orders/:id

HTTP Method: DELETE /orders/:id

## REST API: Sales Orders

URI: /orders

Allows you to retrieve information on existing order/orders.

**URL Structure:** `http://magentohost/api/rest/orders`

**Version:** 1

### HTTP Method: GET /orders

**Description:** Allows you to retrieve the list of existing orders. Each order contains the following information: general order information, information on ordered items, order comments, and order addresses (both billing and shipping).

The list of attributes that will be returned for the order is configured in the Magento Admin Panel.

**Authentication:** Admin, Customer, Guest

**Default Format:** XML**Response Example: XML**

```
GET http://magentohost/api/rest/orders

<?xml version="1.0"?>
<magento_api>
 <data_item_1>
 <customer_id>3</customer_id>
 <base_discount_amount>0.0000</base_discount_amount>
 <base_shipping_amount>455.0000</base_shipping_amount>
 <base_shipping_tax_amount>0.0000</base_shipping_tax_amount>
 <base_subtotal>13650.0000</base_subtotal>
 <base_tax_amount>0.0000</base_tax_amount>
 <base_total_paid></base_total_paid>
 <base_total_refunded></base_total_refunded>
 <tax_amount>0.0000</tax_amount>
 <total_paid></total_paid>
 <total_refunded></total_refunded>
 <base_shipping_discount_amount>0.0000</base_shipping_discount_amount>
 <base_subtotal_incl_tax>13650.0000</base_subtotal_incl_tax>
 <base_total_due>14105.0000</base_total_due>
 <total_due>14105.0000</total_due>
 <base_currency_code>USD</base_currency_code>
 <tax_name></tax_name>
 <tax_rate></tax_rate>
 <addresses>
 <data_item>
 <region>Palau</region>
 <postcode>19103</postcode>
 <lastname>Doe</lastname>
 <street>2356 Jody Road Philadelphia, PA 19103</street>
 <city>PA</city>
 <telephone>610-634-1181</telephone>
 <country_id>US</country_id>
 <firstname>John</firstname>
 <address_type>billing</address_type>
 <prefix></prefix>
 <middlename></middlename>
 <suffix></suffix>
 <company></company>
 </data_item>
 <data_item>
 <region>Palau</region>
 <postcode>19103</postcode>
 <lastname>Doe</lastname>
 <street>2356 Jody Road Philadelphia, PA 19103</street>
 <city>PA</city>
 <telephone>610-634-1181</telephone>
 <country_id>US</country_id>
 <firstname>John</firstname>
 <address_type>shipping</address_type>
 <prefix></prefix>
 <middlename></middlename>
 <suffix></suffix>
 <company></company>
 </data_item>
 </addresses>
 <order_items>
 <data_item>
 <sku>Sunglasses_1</sku>
 <price>150.0000</price>
```

```

<base_price>150.0000</base_price>
<base_original_price>150.0000</base_original_price>
<tax_percent>0.0000</tax_percent>
<tax_amount>0.0000</tax_amount>
<base_tax_amount>0.0000</base_tax_amount>
<base_discount_amount>0.0000</base_discount_amount>
<base_row_total>13650.0000</base_row_total>
<base_price_incl_tax>150.0000</base_price_incl_tax>
<base_row_total_incl_tax>13650.0000</base_row_total_incl_tax>
</data_item>
</order_items>
</data_item_1>
<data_item_2>
<customer_id>3</customer_id>
<base_discount_amount>0.0000</base_discount_amount>
<base_shipping_amount>95.0000</base_shipping_amount>
<base_shipping_tax_amount>0.0000</base_shipping_tax_amount>
<base_subtotal>3350.0000</base_subtotal>
<base_tax_amount>0.0000</base_tax_amount>
<base_total_paid>2445.0000</base_total_paid>
<base_total_refunded>1845.0000</base_total_refunded>
<tax_amount>0.0000</tax_amount>
<total_paid>2445.0000</total_paid>
<total_refunded>1845.0000</total_refunded>
<base_shipping_discount_amount>0.0000</base_shipping_discount_amount>
<base_subtotal_incl_tax>3350.0000</base_subtotal_incl_tax>
<base_total_due>1000.0000</base_total_due>
<total_due>1000.0000</total_due>
<base_currency_code>USD</base_currency_code>
<tax_name></tax_name>
<tax_rate></tax_rate>
<addresses>
<data_item>
<region>Palau</region>
<postcode>19103</postcode>
<lastname>Doe</lastname>
<street>2356 Jody Road Philadelphia, PA 19103</street>
<city>PA</city>
<telephone>610-634-1181</telephone>
<country_id>US</country_id>
<firstname>John</firstname>
<address_type>billing</address_type>
<prefix></prefix>
<middlename></middlename>
<suffix></suffix>
<company></company>
</data_item>
<data_item>
<region>Palau</region>
<postcode>19103</postcode>
<lastname>Doe</lastname>
<street>2356 Jody Road Philadelphia, PA 19103</street>
<city>PA</city>
<telephone>610-634-1181</telephone>
<country_id>US</country_id>
<firstname>John</firstname>
<address_type>shipping</address_type>
<prefix></prefix>
<middlename></middlename>
<suffix></suffix>
<company></company>
</data_item>
</addresses>

```

```

<order_items>
 <data_item>
 <sku>Sunglasses_1</sku>
 <price>150.0000</price>
 <base_price>150.0000</base_price>
 <base_original_price>150.0000</base_original_price>
 <tax_percent>0.0000</tax_percent>
 <tax_amount>0.0000</tax_amount>
 <base_tax_amount>0.0000</base_tax_amount>
 <base_discount_amount>0.0000</base_discount_amount>
 <base_row_total>1350.0000</base_row_total>
 <base_price_incl_tax>150.0000</base_price_incl_tax>
 <base_row_total_incl_tax>1350.0000</base_row_total_incl_tax>
 </data_item>
 <data_item>
 <sku>Sun_glasses</sku>
 <price>200.0000</price>
 <base_price>200.0000</base_price>
 <base_original_price>200.0000</base_original_price>
 <tax_percent>0.0000</tax_percent>
 <tax_amount>0.0000</tax_amount>
 <base_tax_amount>0.0000</base_tax_amount>
 <base_discount_amount>0.0000</base_discount_amount>
 <base_row_total>2000.0000</base_row_total>
 <base_price_incl_tax>200.0000</base_price_incl_tax>
 <base_row_total_incl_tax>2000.0000</base_row_total_incl_tax>
 </data_item>
</order_items>
</data_item_2>
</magento_api>

```

### **HTTP Method: POST /orders**

**Description:** Not implemented.

### **HTTP Method: PUT /orders**

**Description:** Not implemented.

### **HTTP Method: DELETE /orders**

**Description:** Not implemented.

## REST API: Sales Order

URI: /orders/:id

Allows you to retrieve information on a single order.

The list of attributes that will be returned for the order is configured in the Magento Admin Panel.

**URL Structure:** <http://magentohost/api/rest/orders/:id>

**Version:** 1

### **HTTP Method: GET /orders/:id**

**Description:** Allows you to retrieve the order information.

**Authentication:** Admin, Customer

**Default Format:** XML

**Response Example: XML**

```
GET http://magentohost/api/rest/orders/32
```

```
<?xml version="1.0"?>
<magento_api>
 <customer_id>3</customer_id>
 <base_discount_amount>0.0000</base_discount_amount>
 <base_shipping_amount>0.0000</base_shipping_amount>
 <base_shipping_tax_amount>0.0000</base_shipping_tax_amount>
 <base_subtotal>5220.0000</base_subtotal>
 <base_tax_amount>430.6500</base_tax_amount>
 <base_total_paid></base_total_paid>
 <base_total_refunded></base_total_refunded>
 <tax_amount>304.3300</tax_amount>
 <total_paid></total_paid>
 <total_refunded></total_refunded>
 <base_shipping_discount_amount>0.0000</base_shipping_discount_amount>
 <base_subtotal_incl_tax></base_subtotal_incl_tax>
 <base_total_due></base_total_due>
 <total_due></total_due>
 <base_currency_code>USD</base_currency_code>
 <tax_name>US-CA-* -Rate 1</tax_name>
 <tax_rate>8.2500</tax_rate>
 <addresses>
 <data_item>
 <region>Palau</region>
 <postcode>19103</postcode>
 <lastname>Doe</lastname>
 <street>2356 Jody Road Philadelphia
844 Jefferson Street; 4510 Willis Avenue</street>
 <city>PA</city>
 <telephone>610-634-1181</telephone>
 <country_id>US</country_id>
 <firstname>John</firstname>
 <address_type>billing</address_type>
 <prefix>Dr.</prefix>
 <middlename></middlename>
 <suffix>Jr.</suffix>
 <company></company>
 </data_item>
 <data_item>
 <region>Palau</region>
 <postcode>19103</postcode>
 <lastname>Doe</lastname>
 <street>2356 Jody Road Philadelphia
844 Jefferson Street; 4510 Willis Avenue</street>
 <city>PA</city>
 <telephone>610-634-1181</telephone>
 <country_id>US</country_id>
 <firstname>John</firstname>
 <address_type>shipping</address_type>
 <prefix>Dr.</prefix>
 <middlename></middlename>
 <suffix>Jr.</suffix>
 <company></company>
 </data_item>
 </addresses>
 <order_items>
 <data_item>
 <sku>Sun_glasses</sku>
 <price>141.3400</price>
 <base_price>200.0000</base_price>
 <base_original_price>200.0000</base_original_price>
 <tax_percent>8.2500</tax_percent>
```

```

<tax_amount>11.6600</tax_amount>
<base_tax_amount>16.5000</base_tax_amount>
<base_discount_amount>0.0000</base_discount_amount>
<base_row_total>200.0000</base_row_total>
<base_price_incl_tax>216.5000</base_price_incl_tax>
<base_row_total_incl_tax>216.5000</base_row_total_incl_tax>
</data_item>
<data_item>
<sku>Virtual_product</sku>
<price>14.1340</price>
<base_price>20.0000</base_price>
<base_original_price>20.0000</base_original_price>
<tax_percent>8.2500</tax_percent>
<tax_amount>1.1700</tax_amount>
<base_tax_amount>1.6500</base_tax_amount>
<base_discount_amount>0.0000</base_discount_amount>
<base_row_total>20.0000</base_row_total>
<base_price_incl_tax>21.6500</base_price_incl_tax>
<base_row_total_incl_tax>21.6500</base_row_total_incl_tax>
</data_item>
<data_item>
<sku>test_simple_product</sku>
<price>353.3500</price>
<base_price>500.0000</base_price>
<base_original_price>500.0000</base_original_price>
<tax_percent>8.2500</tax_percent>
<tax_amount>291.5000</tax_amount>
<base_tax_amount>412.5000</base_tax_amount>
<base_discount_amount>0.0000</base_discount_amount>
<base_row_total>5000.0000</base_row_total>
<base_price_incl_tax>541.2500</base_price_incl_tax>
<base_row_total_incl_tax>5412.5000</base_row_total_incl_tax>
</data_item>
</order_items>
</magento_api>

```

### **HTTP Method: POST /orders/:id**

**Description:** Not implemented.

### **HTTP Method: PUT /orders/:id**

**Description:** Not implemented.

### **HTTP Method: DELETE /orders/:id**

**Description:** Not implemented.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Order Addresses

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ▲

Inventory

## Orders ▾

Order Addresses

Order Comments

Order Items

## Products ▾

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

### REST API: Order Addresses

URI: /orders/:orderid/addresses

HTTP Method: GET

HTTP Method: POST

HTTP Method: PUT

HTTP Method: DELETE

URI: /orders/:orderid/addresses/billing

HTTP Method: GET

HTTP Method: POST

HTTP Method: PUT

HTTP Method: DELETE

URI: /orders/:orderid/addresses/shipping

HTTP Method: GET

HTTP Method: POST

HTTP Method: PUT

HTTP Method: DELETE

### Order Addresses Attributes

## REST API: Order Addresses

URI: /orders/:orderid/addresses

Allows you to retrieve information about billing and shipping addresses of the required order.

**URL Structure:** <http://magentohost/api/rest/orders/:orderid/addresses>

**Version:** 1

## HTTP Method: GET

**Description:** Allows you to retrieve information on billing and shipping addresses from the required order.

**Notes:** Customers can retrieve addresses only from their orders.

**Authentication:** Admin, Customer

**Default Format:** XML

**Parameters:**

*No Parameters*

### Example:

```
GET http://magentohost/api/rest/orders/32/addresses
```

### Response Body:

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <region>Palau</region>
 <postcode>19103</postcode>
 <lastname>Doe</lastname>
 <street>2356 Jody Road Philadelphia
844 Jefferson Street; 4510 Willis Avenue</street>
 <city>PA</city>
 <telephone>610-634-1181</telephone>
 <country_id>US</country_id>
 <firstname>John</firstname>
 <address_type>billing</address_type>
 <prefix>Dr.</prefix>
 <middlename></middlename>
 <suffix>Jr.</suffix>
 <company></company>
 </data_item>
 <data_item>
 <region>Massachusetts</region>
 <postcode>01852</postcode>
 <lastname>Doe</lastname>
 <street>1073 Smith Street</street>
 <city>Lowell</city>
 <telephone>508-857-6870</telephone>
 <country_id>US</country_id>
 <firstname>John</firstname>
 <address_type>shipping</address_type>
 <prefix></prefix>
 <middlename></middlename>
 <suffix></suffix>
 <company></company>
 </data_item>
</magento_api>
```

## HTTP Method: POST

**Description:** Not allowed.

## HTTP Method: PUT

**Description:** Not allowed.

## HTTP Method: DELETE

**Description:** Not allowed.

URI: /orders/:orderid/addresses/billing

### HTTP Method: GET

**Description:** Allows you to retrieve information on the order billing address.

**Notes:** Customers can retrieve information on billing addresses only from their own orders.

**Authentication:** Admin, Customer

**Default Format:** XML

**Parameters:**

*No Parameters*

#### Example:

```
GET http://magentohost/api/rest/orders/32/addresses/billing
```

#### Response example:

```
<?xml version="1.0"?>
<magento_api>
 <region>Palau</region>
 <postcode>19103</postcode>
 <lastname>Doe</lastname>
 <street>2356 Jody Road Philadelphia
844 Jefferson Street; 4510 Willis Avenue</street>
 <city>PA</city>
 <telephone>610-634-1181</telephone>
 <country_id>US</country_id>
 <firstname>John</firstname>
 <address_type>billing</address_type>
 <prefix>Dr.</prefix>
 <middlename></middlename>
 <suffix>Jr.</suffix>
 <company></company>
</magento_api>
```

### HTTP Method: POST

**Description:** Not allowed.

### HTTP Method: PUT

**Description:** Not allowed.

### HTTP Method: DELETE

**Description:** Not allowed.

URI: /orders/:orderid/addresses/shipping

### HTTP Method: GET

**Description:** Allows you to retrieve information on the order shipping address.

**Notes:** Customers can retrieve information on shipping addresses only from their own orders.

**Authentication:** Admin, Customer

**Default Format:** XML

**Parameters:**

*No Parameters*

**Example:**

```
GET http://magentohost/api/rest/orders/32/addresses/shipping
```

**Response example:**

```
<?xml version="1.0"?>
<magento_api>
 <region>Massachusetts</region>
 <postcode>01852</postcode>
 <lastname>Doe</lastname>
 <street>1073 Smith Street</street>
 <city>Lowell</city>
 <telephone>508-857-6870</telephone>
 <country_id>US</country_id>
 <firstname>John</firstname>
 <address_type>shipping</address_type>
 <prefix></prefix>
 <middlename></middlename>
 <suffix></suffix>
 <company></company>
</magento_api>
```

**HTTP Method: POST**

**Description:** Not allowed.

**HTTP Method: PUT**

**Description:** Not allowed.

**HTTP Method: DELETE**

**Description:** Not allowed.

## Order Addresses Attributes

| Attribute Name       | Attribute Description           |
|----------------------|---------------------------------|
| Customer Last Name   | Customer last name              |
| Customer First Name  | Customer first name             |
| Customer Middle Name | Customer middle name or initial |
| Customer Prefix      | Customer prefix                 |
| Customer Suffix      | Customer suffix                 |
| Company              | Company name                    |
| Street               | Street address                  |
| City                 | City                            |
| State                | State                           |
| ZIP/Postal Code      | ZIP or postal code              |
| Country              | Country name                    |
| Phone Number         | Customer phone number           |

|              |                                                                  |
|--------------|------------------------------------------------------------------|
| Address Type | Address type. Can have the following values: billing or shipping |
|--------------|------------------------------------------------------------------|

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Order Comments

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ▲

Inventory

## Orders ▲

Order Addresses

Order Comments

Order Items

## Products ▾

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

### REST API: Order Comments

URI: /orders/:orderid/comments

HTTP Method: GET

HTTP Method: POST

HTTP Method: PUT

HTTP Method: DELETE

#### Order Comments Attributes

## REST API: Order Comments

URI: /orders/:orderid/comments

Allows you to retrieve information about comments of the required order.

**URL Structure:** `http://magentohost/api/rest/orders/:orderid/comments`

**Version:** 1

### HTTP Method: GET

**Description:** Allows you to retrieve information about comments of the required order.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

*No Parameters*

### Example:

```
GET http://magentohost/api/rest/orders/33/comments
```

### Response Body:

```

<?xml version="1.0"?>
<magento_api>
 <data_item>
 <created_at>2012-03-09 11:20:49</created_at>
 <comment></comment>
 <is_customer_notified>1</is_customer_notified>
 <is_visible_on_front>0</is_visible_on_front>
 <status>pending</status>
 </data_item>
 <data_item>
 <created_at>2012-03-09 11:21:32</created_at>
 <comment>This is a new order for John Doe.</comment>
 <is_customer_notified>1</is_customer_notified>
 <is_visible_on_front>1</is_visible_on_front>
 <status>pending</status>
 </data_item>
</magento_api>

```

**Authentication:** Customer

**Default Format:** XML

**Parameters:**

*No Parameters*

#### **Example:**

```
GET http://magentohost/api/rest/orders/33/comments
```

#### **Response Body:**

```

<?xml version="1.0"?>
<magento_api>
 <data_item>
 <created_at>2012-03-09 11:21:32</created_at>
 <comment>This is a new order for John Doe.</comment>
 </data_item>
</magento_api>

```

#### **HTTP Method: POST**

**Description:** Not allowed.

#### **HTTP Method: PUT**

**Description:** Not allowed.

#### **HTTP Method: DELETE**

**Description:** Not allowed.

## Order Comments Attributes

| Attribute Name | Attribute Description                                                                              | Notes              |
|----------------|----------------------------------------------------------------------------------------------------|--------------------|
| Comment Date   | Date when the comment was added                                                                    | Admin and Customer |
| Comment Text   | Comment text                                                                                       | Admin and Customer |
| Is Customer    | Defines whether the customer is notified about the comment. Can have the following values: 0 or 1. | Admin only         |

|                                      |                                                                                                                                               |            |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|------------|
| Notified                             | values: 0 - Customer is not notified, 1 - Customer is notified.                                                                               |            |
| Is Comment<br>Visible on<br>Frontend | Defines whether the comment is visible on the frontend. Can have the following values: 0<br>- Comment is not visible, 1 - Comment is visible. | Admin only |
| Comment Status                       | Comment status.                                                                                                                               | Admin only |

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Order Items

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ▾

Inventory

### Orders ▾

Order Addresses

Order Comments

[Order Items](#)

### Products ▾

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

### REST API: Order Items

URI: /orders/:orderId/items

HTTP Method: GET

HTTP Method: POST

HTTP Method: PUT

HTTP Method: DELETE

## REST API: Order Items

URI: /orders/:orderId/items

Allows you to retrieve information about order items.

**URL Structure:** `http://magentohost/api/rest/orders/:orderId/items`

**Version:** 1

### HTTP Method: GET

**Description:** Allows you to retrieve the list of existing order items with detailed items information.

**Notes:** The list of attributes that will be returned for order items is configured in the Magento Admin Panel.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

*No Parameters*

### Response Example: XML

```
GET http://magentohost/api/rest/orders/410/items
```

### Response Body:

```

<?xml version="1.0"?>
<magento_api>
 <data_item>
 <item_id>43</item_id>
 <parent_item_id></parent_item_id>
 <sku>Sunglasses_1</sku>
 <name>Sunglasses 1</name>
 <qty_canceled>0.0000</qty_canceled>
 <qty_invoiced>0.0000</qty_invoiced>
 <qty_ordered>3.0000</qty_ordered>
 <qty_refunded>0.0000</qty_refunded>
 <qty_shipped>0.0000</qty_shipped>
 <price>74.9137</price>
 <base_price>106.0050</base_price>
 <original_price>106.0050</original_price>
 <base_original_price>150.0000</base_original_price>
 <tax_percent>8.2500</tax_percent>
 <tax_amount>26.2500</tax_amount>
 <base_tax_amount>37.1400</base_tax_amount>
 <discount_amount>212.0100</discount_amount>
 <base_discount_amount>300.0000</base_discount_amount>
 <row_total>318.0300</row_total>
 <base_row_total>450.0000</base_row_total>
 <price_incl_tax>114.7550</price_incl_tax>
 <base_price_incl_tax>162.3800</base_price_incl_tax>
 <row_total_incl_tax>344.2650</row_total_incl_tax>
 <base_row_total_incl_tax>487.1400</base_row_total_incl_tax>
 <status>Ordered</status>
 </data_item>
 <data_item>
 <item_id>44</item_id>
 <parent_item_id></parent_item_id>
 <sku>test_simple_product</sku>
 <name>test simple product</name>
 <qty_canceled>0.0000</qty_canceled>
 <qty_invoiced>0.0000</qty_invoiced>
 <qty_ordered>10.0000</qty_ordered>
 <qty_refunded>0.0000</qty_refunded>
 <qty_shipped>0.0000</qty_shipped>
 <price>249.7124</price>
 <base_price>353.3500</base_price>
 <original_price>353.3500</original_price>
 <base_original_price>500.0000</base_original_price>
 <tax_percent>8.2500</tax_percent>
 <tax_amount>291.5000</tax_amount>
 <base_tax_amount>412.5000</base_tax_amount>
 <discount_amount>706.7000</discount_amount>
 <base_discount_amount>1000.0000</base_discount_amount>
 <row_total>3533.5000</row_total>
 <base_row_total>5000.0000</base_row_total>
 <price_incl_tax>382.5000</price_incl_tax>
 <base_price_incl_tax>541.2500</base_price_incl_tax>
 <row_total_incl_tax>3825.0000</row_total_incl_tax>
 <base_row_total_incl_tax>5412.5000</base_row_total_incl_tax>
 <status>Ordered</status>
 </data_item>
</magento_api>

```

**Authentication:** Customer

**Default Format:** XML

**Parameters:**

*No Parameters*

## Response Example: XML

```
GET http://magentohost/api/rest/orders/410/items
```

### Response Body:

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <item_id>43</item_id>
 <parent_item_id></parent_item_id>
 <sku>Sunglasses_1</sku>
 <name>Sunglasses 1</name>
 <qty_canceled>0.0000</qty_canceled>
 <qty_invoiced>0.0000</qty_invoiced>
 <qty_ordered>3.0000</qty_ordered>
 <qty_refunded>0.0000</qty_refunded>
 <qty_shipped>0.0000</qty_shipped>
 <price>74.9137</price>
 <row_total>318.0300</row_total>
 <price_incl_tax>114.7550</price_incl_tax>
 <row_total_incl_tax>344.2650</row_total_incl_tax>
 </data_item>
 <data_item>
 <item_id>44</item_id>
 <parent_item_id></parent_item_id>
 <sku>test_simple_product</sku>
 <name>test simple product</name>
 <qty_canceled>0.0000</qty_canceled>
 <qty_invoiced>0.0000</qty_invoiced>
 <qty_ordered>10.0000</qty_ordered>
 <qty_refunded>0.0000</qty_refunded>
 <qty_shipped>0.0000</qty_shipped>
 <price>249.7124</price>
 <row_total>3533.5000</row_total>
 <price_incl_tax>382.5000</price_incl_tax>
 <row_total_incl_tax>3825.0000</row_total_incl_tax>
 </data_item>
</magento_api>
```

### HTTP Method: POST

**Description:** Not allowed.

### HTTP Method: PUT

**Description:** Not allowed.

### HTTP Method: DELETE

**Description:** Not allowed.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Products

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ^

Inventory

## Orders ▾

## Products ^

Product Categories

Product Images

Product Websites

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

### REST API: Products

URI: /products

HTTP Method: GET /products

HTTP Method: GET /products/category\_id=:id

HTTP Method: POST /products

URI: /products/:id

HTTP Method: GET /products/:id

HTTP Method: PUT /products/:id

HTTP Method: DELETE /products/:id

Possible HTTP Status Codes

## REST API: Products

URI: /products

Allows you to retrieve the list of all products with detailed information.

**URL Structure:** `http://magentohost/api/rest/products`

**Version:** 1

### HTTP Method: GET /products

**Description:** Allows you to retrieve the list of all products with detailed information.

**Notes:** The list of attributes that will be returned in the response is configured in the Magento Admin Panel. The list of attributes differs for each type of user.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

*No Parameters*

GET http://magentohost/api/rest/products

#### Response Example: XML

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <entity_id>1</entity_id>
 <type_id>simple</type_id>
 <sku>dress_test</sku>
 <status>1</status>
 <visibility>4</visibility>
 <tax_class_id>2</tax_class_id>
 <weight>1.0000</weight>
 <price>1500.0000</price>
 <special_price>1000.0000</special_price>
 <name>Wedding dress</name>
 <url_key>dress</url_key>
 <country_of_manufacture>AO</country_of_manufacture>
 <msrp_display_actual_price_type>2</msrp_display_actual_price_type>
 <gift_message_available>1</gift_message_available>
 <news_from_date>2012-03-21 00:00:00</news_from_date>
 <news_to_date>2012-03-24 00:00:00</news_to_date>
 <special_from_date>2012-03-21 00:00:00</special_from_date>
 <special_to_date>2012-03-24 00:00:00</special_to_date>
 <description>White wedding dress</description>
 </data_item>
 <data_item>
 <entity_id>2</entity_id>
 <type_id>simple</type_id>
 <sku>black_sunglasses</sku>
 <status>1</status>
 <visibility>4</visibility>
 <tax_class_id>2</tax_class_id>
 <weight>0.2000</weight>
 <price>500.0000</price>
 <special_price>300.0000</special_price>
 <name>Sunglasses</name>
 <url_key>sunglasses</url_key>
 <country_of_manufacture>AR</country_of_manufacture>
 <msrp_display_actual_price_type>2</msrp_display_actual_price_type>
 <gift_message_available></gift_message_available>
 <news_from_date></news_from_date>
 <news_to_date></news_to_date>
 <special_from_date>2012-03-21 00:00:00</special_from_date>
 <special_to_date>2012-03-24 00:00:00</special_to_date>
 <description>Black sunglasses</description>
 </data_item>
</magento_api>
```

#### Response Example: JSON

```
{"1":
{"entity_id":"1","type_id":"simple","sku":"dress_test","status":"1","visibility":"4","ta
dress","url_key":"dress","country_of_manufacture":"AO","msrp_display_actual_price_type":0
3-21 00:00:00","news_to_date":"2012-03-24 00:00:00","special_from_date":"2012-03-21
00:00:00","special_to_date":"2012-03-24 00:00:00","description":"White wedding
dress"},"2":
{"entity_id":"2","type_id":"simple","sku":"black_sunglasses","status":"1","visibility":0
3-21 00:00:00","special_to_date":"2012-03-24 00:00:00","description":"Black
```

```
sunglasses"} }
```

**Authentication:** Customer, Guest

**Default Format:** XML

**Parameters:**

*No Parameters*

```
GET http://magentohost/api/rest/products
```

#### Response Example: XML

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <entity_id>1</entity_id>
 <type_id>simple</type_id>
 <sku>dress_test</sku>
 <description>White wedding dress</description>
 <short_description>dress</short_description>
 <meta_keyword>white, dress</meta_keyword>
 <name>Wedding dress</name>
 <meta_title>dress</meta_title>
 <meta_description>A nice wedding dress</meta_description>
 <regular_price_with_tax>1500</regular_price_with_tax>
 <regular_price_without_tax>1500</regular_price_without_tax>
 <final_price_with_tax>1000</final_price_with_tax>
 <final_price_without_tax>1000</final_price_without_tax>
 <is_saleable>1</is_saleable>

 <image_url>http://magentohost/media/catalog/product/cache/0/image/9df78eab33525d08d6e5fb
 </data_item>
 <data_item>
 <entity_id>2</entity_id>
 <type_id>simple</type_id>
 <sku>black_sunglasses</sku>
 <description>Black sunglasses</description>
 <short_description>Black sunglasses</short_description>
 <meta_keyword>sunglasses, black</meta_keyword>
 <name>Sunglasses</name>
 <meta_title>sunglasses</meta_title>
 <meta_description>black sunglasses</meta_description>
 <regular_price_with_tax>500</regular_price_with_tax>
 <regular_price_without_tax>500</regular_price_without_tax>
 <final_price_with_tax>300</final_price_with_tax>
 <final_price_without_tax>300</final_price_without_tax>
 <is_saleable>1</is_saleable>

 <image_url>http://magentohost/media/catalog/product/cache/0/image/9df78eab33525d08d6e5fb
 1.jpg</image_url>
 </data_item>
 </magento_api>
```

#### Response Example: JSON

```
{"1":{"entity_id":"1","type_id":"simple","sku":"dress_test","description":"White
wedding dress","short_description":"dress","meta_keyword":"white,
dress","name":"Wedding dress","meta_title":"dress","meta_description":"A nice wedding
dress","regular_price_with_tax":1500,"regular_price_without_tax":1500,"final_price_with_
{"entity_id":"2","type_id":"simple","sku":"black_sunglasses","description":"Black
sunglasses","short_description":"Black sunglasses","meta_keyword":"sunglasses,
```

```

 black", "name": "Sunglasses", "meta_title": "sunglasses", "meta_description": "black sunglasses", "regular_price_with_tax": 500, "regular_price_without_tax": 500, "final_price_with_tax": 500, "image": "1.jpg"{}}

```

## HTTP Method: GET /products/category\_id=:id

**Description:** Allows you to retrieve the list of products of a specified category. These products will be returned in the product position ascending order.

In the following example, product with ID=4 has position equal to 7 and the product with ID=3 has position equal to 1. The list of products, therefore, is sorted by the product position in the category.

```
GET http://magentohost/api/rest/products?category_id=5
```

### Response Example: XML

```

<?xml version="1.0"?>
<magento_api>
 <data_item>
 <entity_id>4</entity_id>
 <attribute_set_id>4</attribute_set_id>
 <type_id>simple</type_id>
 <tax_class_id>2</tax_class_id>
 <weight>1.0000</weight>
 <price>329.9900</price>
 </data_item>
 <data_item>
 <entity_id>3</entity_id>
 <attribute_set_id>4</attribute_set_id>
 <type_id>simple</type_id>
 <tax_class_id>2</tax_class_id>
 <weight>1.0000</weight>
 <price>550.0000</price>
 </data_item>
</magento_api>

```

## HTTP Method: POST /products

**Description:** Allows you to create a new simple product.

**Authentication:** Admin

**Default Format:** JSON

**Parameters:**

| Name             | Description                                | Required | Type   | Example Value         |
|------------------|--------------------------------------------|----------|--------|-----------------------|
| type_id          | Product type. Can have the "simple" value. | required | string | simple                |
| attribute_set_id | Attribute set for the product.             | required | int    | 4                     |
| sku              | Product SKU                                | required | string | new_product           |
| name             | Product name                               | required | string | New product           |
| meta_title       | Product meta title                         | optional | string | new product           |
| meta_description | Product meta description                   | optional | string | This is a new product |
| url_key          | A friendly URL path for the product        | optional | string | new-product           |
| custom_design    | Custom design applied for the product page | optional | string | enterprise/default    |

|                                |                                                                                                                                                 |          |        |                        |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------|------------------------|
| page_layout                    | Page template that can be applied to the product page                                                                                           | optional | string | one_column             |
| options_container              | Defines how the custom options for the product will be displayed. Can have the following values: Block after Info Column or Product Info Column | optional | string | container2             |
| country_of_manufacture         | Product country of manufacture                                                                                                                  | optional | string | AD                     |
| msrp_enabled                   | The Apply MAP option. Defines whether the price in the catalog in the frontend is substituted with a Click for price link                       | optional | int    | 1                      |
| msrp_display_actual_price_type | Defines how the price will be displayed in the frontend. Can have the following values: In Cart, Before Order Confirmation, and On Gesture      | optional | int    | 2                      |
| gift_message_available         | Defines whether the gift message is available for the product                                                                                   | optional | int    | 1                      |
| price                          | Product price                                                                                                                                   | required | string | 2000                   |
| special_price                  | Product special price                                                                                                                           | optional | string | 150                    |
| weight                         | Product weight                                                                                                                                  | required | string | 0.5                    |
| msrp                           | The Manufacturer's Suggested Retail Price option. The price that a manufacturer suggests to sell the product at                                 | optional | string | 140                    |
| status                         | Product status. Can have the following values: 1- Enabled, 2 - Disabled.                                                                        | required | int    | 1                      |
| visibility                     | Product visibility. Can have the following values: 1 - Not Visible Individually, 2 - Catalog, 3 - Search, 4 - Catalog, Search.                  | required | int    | 4                      |
| enable_googlecheckout          | Defines whether the product can be purchased with the help of the Google Checkout payment service. Can have the following values: Yes and No    | optional | int    | 1                      |
| tax_class_id                   | Product tax class. Can have the following values: 0 - None, 2 - taxable Goods, 4 - Shipping, etc., depending on created tax classes.            | required | int    | 7                      |
| description                    | Product description.                                                                                                                            | required | string | This is a new product. |
| short_description              | Product short description.                                                                                                                      | required | string | A new product.         |
| meta_keyword                   | Product meta keywords                                                                                                                           | optional | string | new                    |
| custom_layout_update           | An XML block to alter the page layout                                                                                                           | optional | string | XML body               |
| special_from_date              | Date starting from which the special price will be applied to the product                                                                       | optional | string | 2012-03-15 00:00:00    |
| special_to_date                | Date till which the special price will be applied to the product                                                                                | optional | string | 2012-03-15 00:00:00    |

|                    |                                                                                |          |        |                      |
|--------------------|--------------------------------------------------------------------------------|----------|--------|----------------------|
| news_from_date     | Date starting from which the product is promoted as a new product              | optional | string | 2012-03-15 00:00:00  |
| news_to_date       | Date till which the product is promoted as a new product                       | optional | string | 2012-03-15 00:00:00  |
| custom_design_from | Date starting from which the custom design will be applied to the product page | optional | string | 2012-03-15 00:00:00  |
| custom_design_to   | Date till which the custom design will be applied to the product page          | optional | string | 2012-03-15 00:00:00  |
| group_price        | Product group price                                                            | optional | array  | array of group price |
| tier_price         | Product tier price                                                             | optional | array  | array of tier price  |
| stock_data         | Product inventory data                                                         | optional | array  | array of stock data  |

Array of **Group Price** attributes is as follows:

| Name       | Description    | Required | Type   | Example Value |
|------------|----------------|----------|--------|---------------|
| website_id | Website ID     | optional | int    | 0             |
| cust_group | Customer group | optional | int    | 1             |
| price      | Group price    | optional | string | 140           |

Array of **Tier Price** attributes is as follows:

| Name       | Description    | Required | Type   | Example Value |
|------------|----------------|----------|--------|---------------|
| website_id | Website ID     | optional | int    | 0             |
| cust_group | Customer group | optional | int    | 1             |
| price      | Tier price     | optional | string | 140           |
| price_qty  | Price quantity | optional | string | 10            |

Array of **Stock Data** attributes is as follows:

| Name                  | Description                                                                                                    | Required | Type   | Example Value |
|-----------------------|----------------------------------------------------------------------------------------------------------------|----------|--------|---------------|
| qty                   | Quantity of stock items for the current product                                                                | optional | string | 99            |
| min_qty               | Quantity for stock items to become out of stock                                                                | optional | string | 5             |
| use_config_min_qty    | Choose whether the Config settings will be applied for the Qty for Item's Status to Become Out of Stock option | optional | int    | 1             |
| is_qty_decimal        | Choose whether the product can be sold using decimals (e.g., you can buy 2.5 product)                          | optional | int    | 1             |
| backorders            | Defines whether the customer can place the order for products that are out of stock at the moment.             | optional | int    | 0             |
| use_config_backorders | Choose whether the Config settings will be applied for the Backorders option                                   | optional | int    | 1             |

|                             |                                                                                                                          |          |        |    |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|----------|--------|----|
| min_sale_qty                | Minimum number of items in the shopping cart to be sold                                                                  | optional | string | 10 |
| use_config_min_sale_qty     | Choose whether the Config settings will be applied for the Minimum Qty Allowed in Shopping Cart option                   | optional | int    | 1  |
| max_sale_qty                | Maximum number of items in the shopping cart to be sold                                                                  | optional | string | 50 |
| use_config_max_sale_qty     | Choose whether the Config settings will be applied for the Maximum Qty Allowed in Shopping Cart option                   | optional | int    | 1  |
| is_in_stock                 | Defines whether the product is available for selling.                                                                    | optional | int    | 1  |
| notify_stock_qty            | The number of inventory items below which the customer will be notified                                                  | optional | string | 5  |
| use_config_notify_stock_qty | Choose whether the Config settings will be applied for the Notify for Quantity Below option                              | optional | int    | 1  |
| manage_stock                | Choose whether to view and specify the product quantity and availability and whether the product is in stock management. | optional | int    | 1  |
| use_config_manage_stock     | Choose whether the Config settings will be applied for the Manage Stock option                                           | optional | int    | 1  |
| use_config_qty_increments   | Choose whether the Config settings will be applied for the Qty Increments option                                         | optional | int    | 1  |
| qty_increments              | The product quantity increment value                                                                                     | optional | string | 3  |
| use_config_enable_qty_inc   | Choose whether the Config settings will be applied for the Enable Qty Increments option                                  | optional | int    | 1  |
| enable_qty_increments       | Defines whether the customer can add products only in increments to the shopping cart                                    | optional | int    | 0  |
| is_decimal_divided          | Defines whether the stock items can be divided into multiple boxes for shipping                                          | optional | int    | 0  |

POST http://magentohost/api/rest/products

#### Request Example: JSON

```
{
 "custom_design" : "default/blank",
 "custom_design_from" : "02/16/2012",
 "custom_design_to" : "05/01/2012",
 "description" : "Test description",
 "gift_message_available" : 1,
 "meta_description" : "Test meta",
 "meta_keyword" : "Test keyword",
 "meta_title" : "Test title",
 "msrp" : 11.015000000000001,
 "msrp_display_actual_price_type" : 1,
 "msrp_enabled" : 1,
 "name" : "Test",
 "news_from_date" : "02/16/2012",
 "news_to_date" : "16.02.2012",
 "options_container" : "container1",
 "page_layout" : "one_column",
 "price" : 25.5,
 "attribute_set_id" : "4",
 "short_description" : "Test short description",
}
```

```

 "sku" : "simple4f5490f31959f",
 "special_from_date" : "02/16/2012",
 "special_price" : 11.199999999999999,
 "special_to_date" : "03/17/2012",
 "status" : 1,
 "stock_data" : { "backorders" : 1,
 "enable_qty_increments" : 0,
 "is_in_stock" : 0,
 "is_qty_decimal" : 0,
 "manage_stock" : 1,
 "max_sale_qty" : 1,
 "min_qty" : 1.5600000000000001,
 "min_sale_qty" : 1,
 "notify_stock_qty" : -50.990000000000002,
 "qty" : 1,
 "use_config_manage_stock" : 1,
 "use_config_min_qty" : 1,
 "use_config_min_sale_qty": 1,
 "use_config_max_sale_qty" : 1,
 "use_config_backorders": 1,
 "use_config_enable_qty_inc":1,
 "use_config_notify_stock_qty":1 },
 "tax_class_id" : "2",
 "type_id" : "simple",
 "use_config_gift_message_available" : 0,
 "visibility" : "4",
 "weight" : 125
}

```

### Request Example: XML

```

<?xml version="1.0"?>
<magento_api>
 <attribute_set_id>4</attribute_set_id>
 <type_id>simple</type_id>
 <sku>test_dress</sku>
 <name>Test_dress</name>
 <country_of_manufacture>AD</country_of_manufacture>
 <price>2000.0000</price>
 <special_price>1500.0000</special_price>
 <weight>0.5000</weight>
 <status>1</status>
 <visibility>4</visibility>
 <tax_class_id>7</tax_class_id>
 <description>dress</description>
 <short_description>Wedding dress</short_description>
</magento_api>

```

## URI: /products/:id

Allows you to retrieve, update, and delete a specified product in Magento.

**URL Structure:** <http://magentohost/api/rest/products/:id>

**Version:** 1

### HTTP Method: GET /products/:id

**Description:** Allows you to retrieve information on a required simple product.

**Notes:** The list of attributes that will be returned in the response is configured in the Magento Admin Panel. The list of attributes differs for each type of user.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

No Parameters

```
GET http://magentohost/api/rest/products/8
```

**Response Example: XML**

```
<?xml version="1.0"?>
<magento_api>
 <entity_id>8</entity_id>
 <attribute_set_id>4</attribute_set_id>
 <type_id>simple</type_id>
 <sku>dress</sku>
 <name>Dress</name>
 <url_key>my-new-dress</url_key>
 <country_of_manufacture>AD</country_of_manufacture>
 <msrp_display_actual_price_type>2</msrp_display_actual_price_type>
 <gift_message_available>1</gift_message_available>
 <price>2000.0000</price>
 <weight>0.5000</weight>
 <status>1</status>
 <visibility>4</visibility>
 <tax_class_id>7</tax_class_id>
 <description>dress</description>
 <group_price>
 <data_item>
 <website_id>0</website_id>
 <cust_group>1</cust_group>
 <price>1400.0000</price>
 </data_item>
 </group_price>
 <tier_price>
 <data_item>
 <website_id>0</website_id>
 <cust_group>1</cust_group>
 <price>1300.0000</price>
 <price_qty>1.0000</price_qty>
 </data_item>
 </tier_price>
 <stock_data>
 <qty>50.0000</qty>
 <min_qty>0.0000</min_qty>
 <use_config_min_qty>1</use_config_min_qty>
 <is_qty_decimal>0</is_qty_decimal>
 <backorders>0</backorders>
 <use_config_backorders>1</use_config_backorders>
 <min_sale_qty>1.0000</min_sale_qty>
 <use_config_min_sale_qty>1</use_config_min_sale_qty>
 <max_sale_qty>0.0000</max_sale_qty>
 <use_config_max_sale_qty>1</use_config_max_sale_qty>
 <is_in_stock>1</is_in_stock>
 <notify_stock_qty></notify_stock_qty>
 <use_config_notify_stock_qty>1</use_config_notify_stock_qty>
 <manage_stock>0</manage_stock>
 <use_config_manage_stock>1</use_config_manage_stock>
 <use_config_qty_increments>1</use_config_qty_increments>
 <qty_increments>0.0000</qty_increments>
 <use_config_enable_qty_inc>1</use_config_enable_qty_inc>
 <enable_qty_increments>0</enable_qty_increments>
 <is_decimal_divided>0</is_decimal_divided>
 <use config enable qty increments>1</use config enable qty increments>
```

```
</stock_data>
</magento_api>
```

### Response Example: JSON

```
{"entity_id":"8","attribute_set_id":"4","type_id":"simple","sku":"dress","name":"Dress",
new-
dress","country_of_manufacture":"AD","msrp_display_actual_price_type":"2","gift_message_
[{"website_id":0,"cust_group":1,"price":"1400.0000"}],"tier_price":
[{"website_id":0,"cust_group":1,"price":"1300.0000","price_qty":"1.0000"}],"stock_da
{"qty":"50.0000","min_qty":0.0000,"use_config_min_qty":1,"is_qty_decimal":0,"backo
```

**Authentication:** Customer, Guest

**Default Format:** XML

**Parameters:**

*No Parameters*

```
GET http://magentohost/api/rest/products/8
```

### Response Example: XML

```
<?xml version="1.0"?>
<magento_api>
 <entity_id>1</entity_id>
 <type_id>simple</type_id>
 <sku>dress_test</sku>
 <name>Wedding dress</name>
 <meta_title>dress</meta_title>
 <meta_description>A nice wedding dress</meta_description>
 <description>White wedding dress</description>
 <short_description>dress</short_description>
 <meta_keyword>white, dress</meta_keyword>
 <tier_price/>
 <is_in_stock>1</is_in_stock>
 <regular_price_with_tax>1500</regular_price_with_tax>
 <regular_price_without_tax>1500</regular_price_without_tax>
 <final_price_with_tax>1000</final_price_with_tax>
 <final_price_without_tax>1000</final_price_without_tax>
 <is_saleable>1</is_saleable>

<image_url>http://magento/media/catalog/product/cache/0/image/9df78eab33525d08d6e5fb8d27

<url>http://magento/index.php/catalog/product/view/id/1/s/dress/</url>

<buy_now_url>http://magento/index.php/checkout/cart/add/uenc/aHR0cDovLzEyNy4wLjAuMS9Tb3V

 <total_reviews_count>0</total_reviews_count>
 <has_custom_options></has_custom_options>
</magento_api>
```

### Response Example: JSON

```
{"entity_id":1,"type_id":"simple","sku":"dress_test","name":"Wedding
dress","meta_title":"dress","meta_description":"A nice wedding
dress","description":"White wedding
dress","short_description":"dress","meta_keyword":"white, dress","tier_price":
[],"is_in_stock":1,"regular_price_with_tax":1500,"regular_price_without_tax":1500,"fin
```

### HTTP Method: PUT /products/:id

**Description:** Allows you to update an existing product.

**Authentication:** Admin

**Default Format:** JSON

**Parameters:**

*Enter only those parameters which you want to update.*

```
PUT http://magentohost/api/rest/products/8
```

```
PUT for specific store http://magentohost/api/rest/products/8/store/3
```

#### Request Example: JSON

```
{
 "attribute_set_id": "4",
 "type_id": "simple",
 "sku": "wedding dress",
 "name": "Dress_test",
 "meta_title": "dress",
 "meta_description": "a wedding dress",
 "price": "2000.0000",
 "weight": "0.5000",
 "status": "1",
 "visibility": "4",
 "enable_googlecheckout": "1",
 "tax_class_id": "7",
 "description": "dress",
 "meta_keyword": "dress, wedding"
}
```

#### Request Example: XML

```
<?xml version="1.0"?>
<magento_api>
 <attribute_set_id>4</attribute_set_id>
 <type_id>simple</type_id>
 <sku>wedding dress</sku>
 <name>Dress_test</name>
 <meta_title>dress</meta_title>
 <price>2000.0000</price>
 <weight>0.5000</weight>
 <status>1</status>
 <visibility>4</visibility>
 <enable_googlecheckout>1</enable_googlecheckout>
 <tax_class_id>7</tax_class_id>
 <description>dress</description>
 <meta_keyword>dress, wedding</meta_keyword>
</magento_api>
```

#### HTTP Method: DELETE /products/:id

**Description:** Allows you to delete an existing product.

**Authentication:** Admin

**Default Format:** JSON

**Parameters:** no parameters

**Request Example:**

```
DELETE http://magentohost/api/rest/products/1
```

## Possible HTTP Status Codes

| Status Code | Message                              | Description                                                 |
|-------------|--------------------------------------|-------------------------------------------------------------|
| 404         | Resource not found.                  | The required resource is not found.                         |
| 405         | Resource method not implemented yet. | The required method is not implemented yet.                 |
| 405         | Resource does not support method.    | The current resource does not support the specified method. |

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Categories

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ▲

Inventory

## Orders ▾

## Products ▲

[Product Categories](#)

Product Images

Product Websites

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

### REST API: Product Categories

URI: /products/productid/categories

HTTP Method: GET

HTTP Method: POST

URI: /products/productid/categories/categoryid

HTTP Method: DELETE

Possible HTTP Status Codes

## REST API: Product Categories

URI: /products/productid/categories

Allows you to retrieve information about assigned categories, assign, and unassign a category from/to a product.

**URL Structure:** `http://magentohost/api/rest/products/productid/categories`

**Version:** 1

### HTTP Method: GET

**Description:** Allows you to retrieve information about categories assigned to the specified product.

**Authentication:** Admin, Customer

**Default Format:** JSON

**Parameters:**

*No Parameters*

### Example:

```
GET http://magentohost/api/rest/products/8/categories
```

### Response Body:

```
{
 category_id: 8
}
```

### HTTP Method: POST

**Description:** Allows you to assign a category to a specified product.

**Authentication:** Admin

**Default Format:** JSON

**Parameters:**

| Name        | Description     | Required | Type | Example Value |
|-------------|-----------------|----------|------|---------------|
| category_id | The category ID | required | int  | 2             |

### Example:

```
POST http://magentohost/api/rest/products/8/categories
```

### Request Body:

```
{
 "category_id": "2"
}
```

As a result, the category with ID equal to 2 will be assigned to the specified product.

URI: /products/productid/categories/categoryid

### HTTP Method: DELETE

**Description:** Allows you to unassign a category from a specified product.

**Authentication:** Admin

**Default Format:** JSON

**Parameters:**

*No Parameters*

### Example:

```
DELETE http://magentohost/api/rest/products/8/categories/2
```

## Possible HTTP Status Codes

| Status Code | Message                                                            | Description                                                                           |
|-------------|--------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 400         | Product <product ID> is already assigned to category <category ID> | The message is returned when the required category is already assigned to the product |
| 400         | Category not found                                                 | The specified category is not found                                                   |
| 405         | Resource method not implemented yet                                | The specified method is not implemented yet                                           |

**Magento 1.x Reference**

**Magento 1.x Software Support Notice**

This document is valid until June 2020. Starting in your Magento Commerce 1 version, software licensing include both software and security patches. Please review our [Magento Software License Policy](#) to see what's new in your version.

The Magento Open Source 1.9.3.1 & 1.9.4 is providing enhanced security patches through June 2020 to enhance threat vector removal and reduce risk.

**Product Images**

[See the API reference](#) [Edit](#)

**Authentication** [Customer](#) [Guest](#) [HTTP Basic Auth](#) [API Keys](#) [API Filters](#) [Performance Settings](#) [Resources](#) [Delete](#) [Products](#) [Product Categories](#) [Product Types](#) [Product Attributes](#) [Resource Types](#) [Testing REST Resources](#)

**REST API: Product Images**

**URI:** [products/product\\_id/images](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**URIs:** [/products/product\\_id/images](#) [/products/product\\_id/image\\_id](#) [/products/product\\_id/image\\_id/rename](#) [/products/product\\_id/image\\_id/delete](#)

**HTTP Method: GET**

**Description:** Allows you to retrieve information about product images.

**Example:** [GET https://magento1.localhost/index.php/rest/V1/products/1/images](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**Request Body:**

| Name       | Description                                         | Required | Type    | Example Value |
|------------|-----------------------------------------------------|----------|---------|---------------|
| image_id   | The ID of the image to be retrieved.                | yes      | integer | 1             |
| product_id | The ID of the product to which the image belongs.   | yes      | integer | 1             |
| store_id   | The store view ID for which the image is requested. | no       | integer | 1             |

**Note:** If no store is specified or is not defined, the original name is used for the image. The base created image is used if no image is found.

**HTTP Method: POST**

**Description:** Allows you to add a new image for the required product.

**Authentication:** [Customer](#) [Guest](#)

**Parameters:** [All Parameters](#)

**Example:** [POST https://magento1.localhost/index.php/rest/V1/products/1/images](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**Request Body:**

| Name       | Description                                         | Required | Type    | Example Value |
|------------|-----------------------------------------------------|----------|---------|---------------|
| image      | The image file to be uploaded.                      | yes      | binary  | image         |
| product_id | The ID of the product to which the image belongs.   | yes      | integer | 1             |
| store_id   | The store view ID for which the image is requested. | no       | integer | 1             |

**Note:** If no store is specified or is not defined, the original name is used for the image. The base created image is used if no image is found.

**URI:** [products/product\\_id/imageselector/store\\_id](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**URIs:** [/products/product\\_id/imageselector](#) [/products/product\\_id/imageselector/store\\_id](#)

**HTTP Method: GET**

**Description:** Allows you to retrieve information about product images for a specific store view.

**Example:** [GET https://magento1.localhost/index.php/rest/V1/products/1/imageselector/1](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**Request Body:**

| Name       | Description                                         | Required | Type    | Example Value |
|------------|-----------------------------------------------------|----------|---------|---------------|
| product_id | The ID of the product to which the image belongs.   | yes      | integer | 1             |
| store_id   | The store view ID for which the image is requested. | yes      | integer | 1             |

**Note:** If no store is specified or is not defined, the original name is used for the image. The base created image is used if no image is found.

**HTTP Method: POST**

**Description:** Allows you to add an image for the required product with image settings for a specific store view.

**Example:** [POST https://magento1.localhost/index.php/rest/V1/products/1/imageselector/1](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**Request Body:**

| Name        | Description                                         | Required | Type    | Example Value |
|-------------|-----------------------------------------------------|----------|---------|---------------|
| image       | The image file to be uploaded.                      | yes      | binary  | image         |
| product_id  | The ID of the product to which the image belongs.   | yes      | integer | 1             |
| store_id    | The store view ID for which the image is requested. | yes      | integer | 1             |
| image_label | The name for the image.                             | yes      | string  | image         |

**Note:** If no store is specified or is not defined, the original name is used for the image. The base created image is used if no image is found.

**URI:** [products/product\\_id/images/image\\_id](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**URIs:** [/products/product\\_id/images](#) [/products/product\\_id/images/image\\_id](#)

**HTTP Method: GET**

**Description:** Allows you to retrieve information about a specific product image.

**Example:** [GET https://magento1.localhost/index.php/rest/V1/products/1/images/1](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**Request Body:**

| Name     | Description                          | Required | Type    | Example Value |
|----------|--------------------------------------|----------|---------|---------------|
| image_id | The ID of the image to be retrieved. | yes      | integer | 1             |

**Note:** If no store is specified or is not defined, the original name is used for the image. The base created image is used if no image is found.

**HTTP Method: PUT**

**Description:** Allows you to update information for the specified product image.

**Example:** [PUT https://magento1.localhost/index.php/rest/V1/products/1/images/1](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**Request Body:**

| Name        | Description                        | Required | Type    | Example Value |
|-------------|------------------------------------|----------|---------|---------------|
| image_id    | The ID of the image to be updated. | yes      | integer | 1             |
| image_label | The name for the image.            | yes      | string  | image         |
| image       | The image file to be uploaded.     | yes      | binary  | image         |

**Note:** If no store is specified or is not defined, the original name is used for the image. The base created image is used if no image is found.

**HTTP Method: DELETE**

**Description:** Deletes the specified product image.

**Example:** [DELETE https://magento1.localhost/index.php/rest/V1/products/1/images/1](#)

**HTTP Headers:** [Content-Type](#) [Accept](#) [Authorization](#)

**Request Body:**

| Name     | Description                        | Required | Type    | Example Value |
|----------|------------------------------------|----------|---------|---------------|
| image_id | The ID of the image to be deleted. | yes      | integer | 1             |

**Note:** If no store is specified or is not defined, the original name is used for the image. The base created image is used if no image is found.

|             |                                                                             |        |        |        |
|-------------|-----------------------------------------------------------------------------|--------|--------|--------|
| Image       | A brief text of the download or the download of an image file.              | string | string | file   |
| Description | The full URL address. The user is in the store where the image was deleted. | string | string | url    |
| Code        | Any error message. Can leave the following when image: string               | string | string | object |

Example

[\[API Reference\]\(https://docs.mercadopago.com/reference/api-endpoints/images/delete\)](#)**Request Body**

```
HTTP/1.1 POST https://api.mercadopago.com/v1/images/{image_id}
Content-Type: application/json
Authorization: Bearer {token}
{
 "image": "https://www.mercadopago.com.br/images/logo_mercado_pago.png"
}
```

**HTTP Method DELETE****Description:** Allows you to remove the specified image from a product.**Body:** The URL of the image to be removed completely. All image parameters will be set to the image.**Authentication:** Token**Parameters:****Code:** Error code

Example

[\[API Reference\]\(https://docs.mercadopago.com/reference/api-endpoints/images/delete\)](#)**URL:** products/{product\_id}/images/{image\_id}/{store\_id}**HTTP Response:** 200 OK - The image has been successfully deleted.**Variables:** {product\_id} - The ID of the product. {image\_id} - The ID of the image. {store\_id} - The store ID.**HTTP Method PUT****Description:** Allows you to update information about the specified product image from a selected store.**Body:** The URL of the image to be updated completely. All image parameters will be set to the image.**Authentication:** Token**Parameters:****Code:** Error code

Example

[\[API Reference\]\(https://docs.mercadopago.com/reference/api-endpoints/images/update\)](#)**Response Body:**

```
HTTP/1.1 200 OK
Content-Type: application/json
{
 "image": "https://www.mercadopago.com.br/images/logo_mercado_pago.png"
}
```

**HTTP Method PUT****Description:** Allows you to update specific image information for a selected store.**Body:** The URL of the image to be updated completely. All image parameters will be set to the image.**Authentication:** Token**Parameters:****Code:** Error code

Example

[\[API Reference\]\(https://docs.mercadopago.com/reference/api-endpoints/images/update\)](#)**Request Body:**

```
HTTP/1.1 POST https://api.mercadopago.com/v1/images/{image_id}
Content-Type: application/json
Authorization: Bearer {token}
{
 "image": "https://www.mercadopago.com.br/images/logo_mercado_pago.png",
 "store_id": "12345678901234567890123456789012"
}
```

**HTTP Method DELETE****Description:** Allows you to remove an image from the desired product in the specified store.**Body:** The URL of the image to be removed completely. All image parameters will be set to the image.**Authentication:** Token**Parameters:****Code:** Error code

Example

[\[API Reference\]\(https://docs.mercadopago.com/reference/api-endpoints/images/delete\)](#)**Request Body:**

```
HTTP/1.1 POST https://api.mercadopago.com/v1/images/{image_id}
Content-Type: application/json
Authorization: Bearer {token}
{
 "image": "https://www.mercadopago.com.br/images/logo_mercado_pago.png",
 "store_id": "12345678901234567890123456789012"
}
```

**HTTP Method GET****Description:** Allows you to get the image information for a selected store.**Body:** The URL of the image to be updated completely. All image parameters will be set to the image.**Authentication:** Token**Parameters:****Code:** Error code

Example

[\[API Reference\]\(https://docs.mercadopago.com/reference/api-endpoints/images/get\)](#)**Response Body:**

```
HTTP/1.1 200 OK
Content-Type: application/json
{
 "image": "https://www.mercadopago.com.br/images/logo_mercado_pago.png"
}
```

**HTTP Method GET****Description:** Allows you to get the image information for a selected store.**Body:** The URL of the image to be updated completely. All image parameters will be set to the image.**Authentication:** Token**Parameters:****Code:** Error code

Example

[\[API Reference\]\(https://docs.mercadopago.com/reference/api-endpoints/images/get\)](#)**Response Body:**

```
HTTP/1.1 200 OK
Content-Type: application/json
{
 "image": "https://www.mercadopago.com.br/images/logo_mercado_pago.png"
}
```

**HTTP Status Codes****Status:** Message - Description**200:** Request successful - The operation was successful.**201:** Request successful - The specified object was created.**204:** Request successful - The specified object was deleted.**302:** Request successful - Redirection to another page.**400:** The request contains invalid data - The image file content must be less than 10 MB, avoided frame, extension and file size.**401:** Request failed due to authentication errors - The user does not have the required permissions.**403:** Request failed due to restrictions - The user does not have the required permissions.**404:** Request failed due to restrictions - The image file was not found.**405:** Request failed due to restrictions - Response status error.**406:** Request failed due to restrictions - Response content error.**408:** Request time exceeded - The request took too long.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Websites

[Edit this page on GitHub](#) 

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ▾

Inventory

## Orders ▾

## Products ▾

Product Categories

Product Images

[Product Websites](#)

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

### REST API: Product Websites

URI: /products/:product\_id/websites

HTTP Method: GET

HTTP Method: POST

Website Assignment

Website Assignment with Product Data Copying

Multi-Website Assignment

Multi-Website Assignment with Product Data Copying

URI: /products/:product\_id/websites/:website\_id

HTTP Method: DELETE

Possible HTTP Status Codes

## REST API: Product Websites

URI: /products/:product\_id/websites

Allows you to retrieve information about websites assigned to a product, assign a website to a product, and copy data for a product from a specified store view.

**URL Structure:** [http://magentohost/api/rest/products/:product\\_id/websites](http://magentohost/api/rest/products/:product_id/websites)

**Version:** 1

### HTTP Method: GET

**Description:** Allows you to retrieve information about websites assigned to the specified product.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

*No Parameters*

**Example:**

```
GET http://magentohost/api/rest/products/8/websites
```

**Response Body:**

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <website_id>2</website_id>
 </data_item>
</magento_api>
```

**HTTP Method: POST****Website Assignment**

**Description:** Allows you to assign a website to a specified product.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

| Name       | Description    | Required | Type | Example Value |
|------------|----------------|----------|------|---------------|
| website_id | The website ID | required | int  | 2             |

**Example:**

```
POST http://magentohost/api/rest/products/8/websites
```

**Request Body:**

```
<?xml version="1.0"?>
<magento_api>
 <website_id>1</website_id>
</magento_api>
```

**Response Body:**

```
<?xml version="1.0"?>
<magento_api>
 <success>
 <data_item>
 <website_id>1</website_id>
 <product_id>8</product_id>
 <message>Resource updated successful.</message>
 <code>200</code>
 </data_item>
 </success>
</magento_api>
```

**Website Assignment with Product Data Copying**

**Description:** Allows you to assign a website and copy product data from the attached store to the one being

attached. Only product data that is set on the Store View level is copied. All other data set on the Website or Global levels is not copied.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

| Name       | Description                                 | Required | Type | Example Value |
|------------|---------------------------------------------|----------|------|---------------|
| website_id | The website ID                              | required | int  | 2             |
| store_from | The store ID from which data will be copied | required | int  | 1             |
| store_to   | The store ID to which data will be copied   | required | int  | 2             |

**Notes:** The store\_to parameter must belong to the website which we want to assign to a product.

**Example:**

```
POST http://magentohost/api/rest/products/8/websites
```

**Request Body:**

```
<?xml version="1.0"?>
<magento_api>
 <website_id>2</website_id>
 <copy_to_stores>
 <data_item>
 <store_from>1</store_from>
 <store_to>2</store_to>
 </data_item>
 </copy_to_stores>
</magento_api>
```

## Multi-Website Assignment

**Description:** Allows you to assign multiple websites to a product.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

| Name       | Description    | Required | Type | Example Value |
|------------|----------------|----------|------|---------------|
| website_id | The website ID | required | int  | 2             |

**Example:**

```
POST http://magentohost/api/rest/products/8/websites
```

**Request Body:**

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <website_id>1</website_id>
 </data_item>
 <data_item>
```

```

<website_id>3</website_id>
</data_item>
</magento_api>

```

### Response Body:

```

<?xml version="1.0"?>
<magento_api>
<success>
<data_item>
<website_id>1</website_id>
<product_id>8</product_id>
<message>Resource updated successful.</message>
<code>200</code>
</data_item>
<data_item>
<website_id>3</website_id>
<product_id>8</product_id>
<message>Resource updated successful.</message>
<code>200</code>
</data_item>
</success>
</magento_api>

```

### Multi-Website Assignment with Product Data Copying

**Description:** Allows you to assign multiple websites to a product together with copying product data from the attached store to the one being attached. Only product data that is set on the Store View level is copied. All other data set on the Website or Global levels is not copied.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

| Name       | Description                                 | Required | Type | Example Value |
|------------|---------------------------------------------|----------|------|---------------|
| website_id | The website ID                              | required | int  | 2             |
| store_from | The store ID from which data will be copied | required | int  | 1             |
| store_to   | The store ID to which data will be copied   | required | int  | 2             |

### Example:

```
POST http://magentohost/api/rest/products/8/websites
```

### Request Body:

```

<?xml version="1.0"?>
<magento_api>
<data_item>
<website_id>2</website_id>
<copy_to_stores>
<data_item>
<store_from>1</store_from>
<store_to>2</store_to>
</data_item>
</copy_to_stores>

```

```

</data_item>
<data_item>
 <website_id>3</website_id>
 <copy_to_stores>
 <data_item>
 <store_from>1</store_from>
 <store_to>5</store_to>
 </data_item>
 </copy_to_stores>
</data_item>
</magento_api>

```

#### Response Body:

```

<?xml version="1.0"?>
<magento_api>
 <success>
 <data_item>
 <website_id>2</website_id>
 <product_id>8</product_id>
 <message>Resource updated successful.</message>
 <code>200</code>
 </data_item>
 <data_item>
 <website_id>3</website_id>
 <product_id>8</product_id>
 <message>Resource updated successful.</message>
 <code>200</code>
 </data_item>
 </success>
</magento_api>

```

URI: /products/:product\_id/websites/:website\_id

Allows you to unassign a website from a specified product.

**URL Structure:** http://magentohost/api/rest/products/:product\_id/websites/:website\_id

**Version:** 1

#### HTTP Method: DELETE

**Description:** Allows you to unassign a website from a specified product.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

*No Parameters*

#### Example:

```
DELETE http://magentohost/api/rest/products/8/websites/1
```

## Possible HTTP Status Codes

| Status Code | Message | Description |
|-------------|---------|-------------|
|             |         |             |

|     |                                                                                                                    |                                                       |
|-----|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| 404 | Product not found                                                                                                  | The specified product is not found or does not exist. |
| 404 | Website not found                                                                                                  | The specified website is not found or does not exist. |
| 400 | Invalid value for "store_from" for the website with ID <ID value>.                                                 | The entered value for "store_from" is not valid.      |
| 400 | Invalid value for "store_to" for the website with ID <ID value>.                                                   | The entered value for "store_to" is not valid.        |
| 400 | Store not found <store ID> for website <website ID>.                                                               | The specified store is not found or does not exist.   |
| 400 | Store <store ID> from which we will copy the information does not belong to the product <product ID> being edited. | The specified store is not assigned to the product.   |
| 400 | Store <store ID> to which we will copy the information does not belong to the website <website ID> being added.    | The specified store does not belong to the website.   |
| 400 | Product <product ID> isn't assigned to website <website ID>.                                                       | The specified product is not assigned to the website. |
| 400 | Invalid value for "website_id" in request.                                                                         | The value for "website_id" is not valid.              |

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Customer Addresses

[Edit this page on GitHub](#) 

JSON responses on this page contributed by Tim Reynolds

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ▾

Inventory

## Orders ▾

## Products ▾

[Customer Addresses](#)

Customers

## Response Formats

## Testing REST Resources

### REST API: Customer Addresses

URI: /customers/:customer\_id/addresses

HTTP Method: GET /customers/:customer\_id/addresses

HTTP Method: POST /customers/:customer\_id/addresses

HTTP Method: PUT /customers/:customer\_id/addresses

HTTP Method: DELETE /customers/:customer\_id/addresses

URI: /customers/addresses/:address\_id

HTTP Method: GET /customers/addresses/:address\_id

HTTP Method: POST /customers/addresses/:address\_id

HTTP Method: PUT /customers/addresses/:address\_id

HTTP Method: DELETE /customers/addresses/:address\_id

Possible HTTP Status Codes

## REST API: Customer Addresses

URI: /customers/:customer\_id/addresses

Allows you to manage existing customer addresses.

**URL Structure:** [http://magentohost/api/rest/customers/:customer\\_id/addresses](http://magentohost/api/rest/customers/:customer_id/addresses)

**Version:** 1

### HTTP Method: GET /customers/:customer\_id/addresses

**Description:** Allows you to retrieve the list of existing customer addresses.

**Notes:** The list of attributes that will be returned for customer addresses is configured in the Magento Admin Panel. The Customer user type has access only to his/her own addresses. Also, Admin can add additional non-system customer address attributes by selecting **Customers > Attributes > Manage Customer Address**

**Attributes** (available only in Magento Enterprise Edition). If these attributes are set as visible on frontend, they will be returned in the response.

**Authentication:** Admin, Customer

**Default Format:** XML

**Parameters:**

*No Parameters*

**Example:**

```
GET http://magentohost/api/rest/customers/1/addresses
```

**Response Body:**

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <entity_id>2</entity_id>
 <firstname>John</firstname>
 <lastname>Doe</lastname>
 <city>PA</city>
 <region>Palau</region>
 <postcode>19103</postcode>
 <country_id>US</country_id>
 <telephone>610-634-1181</telephone>
 <prefix>Dr.</prefix>
 <middlename></middlename>
 <suffix>Jr.</suffix>
 <company></company>
 <fax></fax>
 <vat_id>123456789</vat_id>
 <street>
 <data_item>2356 Jody Road Philadelphia</data_item>
 <data_item>844 Jefferson Street; 4510 Willis Avenue</data_item>
 </street>
 <is_default_billing>1</is_default_billing>
 <is_default_shipping>1</is_default_shipping>
 </data_item>
</magento_api>
```

**response body: json:**

```
[
 {
 "entity_id": "98",
 "firstname": "John",
 "middlename": null,
 "lastname": "Doe",
 "company": null,
 "city": "Philly",
 "country_id": "US",
 "region": "California",
 "postcode": "94131",
 "telephone": "888-555-1212",
 "fax": null,
 "street": [
 "123 Center St"
],
 "is_default_billing": 0,
 "is_default_shipping": 1
 },
```

```
{
 "entity_id": "99",
 "firstname": "John",
 "lastname": "Doe",
 "city": "San Jose",
 "country_id": "US",
 "region": "California",
 "postcode": "94070",
 "telephone": "898-555-1212",
 "street": [
 "123 Easy St"
],
 "is_default_billing": 1,
 "is_default_shipping": 0
}
]
```

**Notes:** If the customer has more than two street addresses, they will be returned in the following form: first address in the first string and all other addresses in the second string separated with a semicolon (like in the example above).

### HTTP Method: POST /customers/:customer\_id/addresses

**Description:** Allows you to create a new address for the required customer.

**Notes:** The Customer user type can create addresses only for themselves.

When adding a street address for the customer, it should look like the following:

```
<street>
 <data_item>street address 1</data_item>
 <data_item>street address 2</data_item>
 <data_item>street address 3</data_item>
</street>
```

**Authentication:** Admin, Customer

**Default Format:** XML

#### Parameters:

| Name       | Description                                                         | Required                                                | Type   | Example Value    |
|------------|---------------------------------------------------------------------|---------------------------------------------------------|--------|------------------|
| firstname  | Customer first name                                                 | required                                                | string | John             |
| lastname   | Customer last name                                                  | required                                                | string | Doe              |
| street     | Customer street address. There can be more than one street address. | required                                                | string | 2800 Watson Lane |
| city       | Name of the city                                                    | required                                                | string | Asheville        |
| country_id | Name of the country                                                 | required                                                | string | US               |
| region     | Region name or code                                                 | required for countries with regions (USA, Canada, etc.) | string | Palau            |
| postcode   | Customer ZIP/postal code                                            | required                                                | string | 28803            |
| telephone  | Customer phone number                                               | required                                                | string | 828-775-         |

**Example:**

```
POST http://magentohost/api/rest/customers/1/addresses
```

**Request Body:**

```
<?xml version="1.0"?>
<magento_api>
 <firstname>Johny</firstname>
 <lastname>Doe</lastname>
 <city>PA</city>
 <region>Palau</region>
 <postcode>19103</postcode>
 <country_id>US</country_id>
 <telephone>611-634-1181</telephone>
 <street>
 <data_item>2354 Jody Road Philadelphia</data_item>
 <data_item>844 Jefferson Street; 4510 Willis Avenue</data_item>
 </street>
</magento_api>
```

**HTTP Method: PUT /customers/:customer\_id/addresses**

**Update Customer Address:** not allowed.

**HTTP Method: DELETE /customers/:customer\_id/addresses**

**Description:** Not allowed.

**URI:** /customers/addresses/:address\_id

Allows you to manage an existing customer address.

**URL Structure:** http://magentohost/api/rest/customers/addresses/:address\_id

**Version:** 1

**HTTP Method: GET /customers/addresses/:address\_id**

**Description:** Allows you to retrieve an existing customer address.

**Notes:** The list of attributes that will be returned for customer address is configured in the Magento Admin Panel. The Customer user type has access only to his/her own addresses. Also, Admin can add additional non-system customer address attributes by selecting **Customers > Attributes > Manage Customer Address**

**Attributes** (available only in Magento Enterprise Edition). If these attributes are set as visible on frontend, they will be returned in the response.

**Authentication:** Admin, Customer

**Default Format:** XML

**Parameters:**

*No Parameters*

**Example:**

```
GET http://magentohost/api/rest/customers/addresses/2
```

**Response Body:**

```
<?xml version="1.0"?>
<magento_api>
 <entity_id>2</entity_id>
```

```

<firstname>John</firstname>
<lastname>Doe</lastname>
<city>PA</city>
<region>Palau</region>
<postcode>19103</postcode>
<country_id>US</country_id>
<telephone>610-634-1181</telephone>
<prefix>Dr.</prefix>
<middlename></middlename>
<suffix>Jr.</suffix>
<company></company>
<fax></fax>
<vat_id>123456789</vat_id>
<street>
 <data_item>2356 Jody Road Philadelphia</data_item>
 <data_item>844 Jefferson Street; 4510 Willis Avenue</data_item>
</street>
<is_default_billing>1</is_default_billing>
<is_default_shipping>1</is_default_shipping>
</magento_api>

```

**response body: json:**

```
{
 "entity_id": "2",
 "firstname": "John",
 "lastname": "Doe",
 "city": "San Jose",
 "country_id": "US",
 "region": "California",
 "postcode": "94070",
 "telephone": "898-555-1212",
 "street": [
 "123 Easy St"
],
 "is_default_billing": 1,
 "is_default_shipping": 0
}
```

### HTTP Method: POST /customers/addresses/:address\_id

**Description:** Not allowed.

### HTTP Method: PUT /customers/addresses/:address\_id

**Description:** Allows you to update an existing customer address.

**Notes:** The list of attributes that will be updated for customer address is configured in the Magento Admin Panel. The Customer user type has access only to his/her own addresses.

If you want to add more addresses, it should look like the following:

```

<street>
 <data_item>street address 1</data_item>
 <data_item>street address 2</data_item>
 <data_item>street address 3</data_item>
</street>
```

**Authentication:** Admin, Customer

**Default Format:** XML

**Example:**

PUT http://magentohost/api/rest/customers/addresses/7

**Request Body:**

```
<?xml version="1.0"?>
<magento_api>
 <firstname>Johny</firstname>
 <lastname>Doe</lastname>
 <city>PA</city>
 <region>Palau</region>
 <postcode>19103</postcode>
 <country_id>US</country_id>
 <telephone>610-634-1181</telephone>
 <street>
 <data_item>2356 Jody Road Philadelphia</data_item>
 <data_item>844 Jefferson Street</data_item>
 </street>
 <is_default_billing>1</is_default_billing>
 <is_default_shipping>1</is_default_shipping>
</magento_api>
```

**request body: json:**

```
{
 "entity_id": "99",
 "firstname": "John",
 "lastname": "Doe",
 "city": "San Jose",
 "country_id": "US",
 "region": "California",
 "postcode": "94070",
 "telephone": "898-555-1212",
 "street": [
 "123 Easy St"
],
 "is_default_billing": 1,
 "is_default_shipping": 0
}
```

**HTTP Method: DELETE /customers/addresses/:address\_id****Description:** Allows you to delete an existing customer address.**Notes:** The Customer user type can delete only his/her own addresses.**Authentication:** Admin, Customer**Default Format:** XML**Parameters:***No Parameters***Example:**

DELETE http://magentohost/api/rest/customers/addresses/7

## Possible HTTP Status Codes

| Status | Message | Description |
|--------|---------|-------------|
|--------|---------|-------------|

| Code |                                                                 |                                                                                                    |
|------|-----------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| 405  | Resource method not implemented yet.                            | The required method is not implemented or not allowed.                                             |
| 404  | Resource not found.                                             | The specified resource is not found or does not exist.                                             |
| 400  | Invalid country identifier type                                 | The <country_id> value is passed not as a string type.                                             |
| 400  | <value name> is a required value.                               | The specified value is a required one.                                                             |
| 400  | Country does not exist                                          | The specified country does not exist.                                                              |
| 400  | Country is required                                             | The <country_id> value is required.                                                                |
| 400  | Country is not between '2' and '3' inclusively.                 | The country code is not in the range of 2 and 3 symbols inclusively.                               |
| 400  | Invalid State/Province type                                     | The <region> value is invalid (value is empty or passed as an array)                               |
| 400  | State/Province is required                                      | The <region> value is required for the specified country.                                          |
| 400  | State/Province is invalid                                       | The entered value for <region> is invalid. It must be the region code (TX) or region name (Texas). |
| 400  | State/Province does not exist                                   | The specified region does not exist (only for the country with predefined regions).                |
| 400  | Address is default for customer so is not allowed to be deleted | The address cannot be deleted because it is set as a default one for billing or shipping.          |

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Customers

[Edit this page on GitHub](#) 

JSON responses on this page contributed by Tim Reynolds

## Introduction

## Authentication ▾

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings ▾

## Resources ▾

Inventory

## Orders ▾

## Products ▾

Customer Addresses

Customers

## Response Formats

## Testing REST Resources

### REST API: Customers

URI: /customers

HTTP Method: GET /customers

HTTP Method: POST /customers

HTTP Method: PUT /customers

HTTP Method: DELETE /customers

### REST API: Customer

URI: /customers/:customerId

HTTP Method: GET /customers/:customerId

HTTP Method: POST /customers/:customerId

HTTP Method: PUT /customers/:customerId

HTTP Method: DELETE /customers/:customerId

## REST API: Customers

URI: /customers

Allows you to create and retrieve customers.

**URL Structure:** `http://magentohost/api/rest/customers`

**Version:** 1

### HTTP Method: GET /customers

**Description:** Allows you to retrieve the list of existing customers.

**Notes:** Only Admin user can retrieve the list of customers with all their attributes.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

*No Parameters*

### Example

```
GET http://magentohost/api/rest/customers
```

#### Response Body:

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <entity_id>2</entity_id>
 <website_id>1</website_id>
 <email>test@example.com</email>
 <group_id>1</group_id>
 <created_at>2012-03-22 14:15:54</created_at>
 <disable_auto_group_change>1</disable_auto_group_change>
 <firstname>john</firstname>
 <lastname>Doe</lastname>
 <created_in>Default Store View</created_in>
 </data_item>
 <data_item>
 <entity_id>4</entity_id>
 <website_id>1</website_id>
 <email>earl@example.com</email>
 <group_id>1</group_id>
 <created_at>2012-03-28 13:54:04</created_at>
 <disable_auto_group_change>0</disable_auto_group_change>
 <firstname>Earl</firstname>
 <lastname>Hickey</lastname>
 <created_in>Admin</created_in>
 </data_item>
</magento_api>
```

#### response example: json

```
get http://magentohost/api/rest/customers
```

#### response body:

```
{
 "2": {
 "entity_id": "2",
 "website_id": "1",
 "email": "test@example.com",
 "group_id": "1",
 "created_at": "2012-03-22 14:15:54",
 "disable_auto_group_change": "1",
 "firstname": "john",
 "lastname": "Doe",
 "created_in": "Admin",
 "prefix": null,
 "suffix": null,
 "taxvat": null,
 "dob": "2001-01-03 00:00:00",
 "reward_update_notification": "1",
 "reward_warning_notification": "1",
 "gender": "1"
 }
}
```

```

},
"4": {
 "entity_id": "4",
 "website_id": "1",
 "email": "earl@example.com",
 "group_id": "1",
 "created_at": "2013-03-28 18:59:41",
 "disable_auto_group_change": "0",
 "firstname": "Earl",
 "lastname": "Hickey",
 "created_in": "Admin",
 "prefix": null,
 "suffix": null,
 "taxvat": null,
 "dob": "2012-03-28 13:54:04",
 "reward_update_notification": "1",
 "reward_warning_notification": "1",
 "gender": "1"
}
}

```

## HTTP Method: POST /customers

**Description:** Allows you to create a new customer.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

| Name                      | Description                                                                  | Required | Type   | Example Value     |
|---------------------------|------------------------------------------------------------------------------|----------|--------|-------------------|
| firstname                 | The customer first name                                                      | required | string | John              |
| lastname                  | The customer last name                                                       | required | string | Doe               |
| email                     | The customer email address                                                   | required | string | johny@example.com |
| password                  | The customer password. The password must contain minimum 7 characters        | required | string | 123123q           |
| website_id                | Website ID                                                                   | required | int    | 1                 |
| group_id                  | Customer group ID                                                            | required | int    | 1                 |
| disable_auto_group_change | Defines whether the automatic group change for the customer will be disabled | optional | int    | 0                 |
| prefix                    | Customer prefix                                                              | optional | string | Mr.               |
| middlename                | Customer middle name or initial                                              | optional | string | R.                |
| suffix                    | Customer suffix                                                              | optional | string | Sr.               |
| taxvat                    | Customer Tax or VAT number                                                   | optional | string | GB999 9999 73     |

**Notes:** The list of parameters may change depending on the attributes settings in **Customers > Attributes >**

**Manage Customer Attributes** page in Magento Admin Panel. For example, a required status of the **middlename** attribute (Middle Name/Initial) may be changed to 'YES'. Please note that managing customer attributes is available only in Magento Enterprise Edition.

**Example:**

```
POST http://magentohost/api/rest/customers
```

### Request Body:

```
<?xml version="1.0"?>
<magento_api>
 <firstname>Earl</firstname>
 <lastname>Hickey</lastname>
 <password>123123q</password>
 <email>earl@example.com</email>
 <website_id>1</website_id>
 <group_id>1</group_id>
</magento_api>
```

### Response:

If the customer was created successfully, we receive **Response HTTP Code** = 200, empty **Response Body** and **Location** header like '/api/rest/customers/555' where '555' - an entity id of the new customer.

### HTTP Method: PUT /customers

**Description:** Not allowed

### HTTP Method: DELETE /customers

**Description:** Not allowed

## REST API: Customer

URI: /customers/:customerId

Allows you to manage existing customers.

**URL Structure:** http://magentohost/api/rest/customers/:customerId

**Version:** 1

### HTTP Method: GET /customers/:customerId

**Description:** Allows you to retrieve information on an existing customer.

**Notes:** The list of attributes that will be returned for customers is configured in the Magento Admin Panel. The Customer user type has access only to his/her own information. Also, Admin can add additional non-system customer attributes by selecting **Customers > Attributes > Manage Customer Attributes**. If these attributes are set as visible on frontend, they will be returned in the response. Also, custom attributes will be returned in the response only after the customer information is updated in the Magento Admin Panel or the specified custom attribute is updated via API (see the PUT method below). Please note that managing customer attributes is available only in Magento Enterprise Edition.

**Authentication:** Admin, Customer

**Default Format:** XML

**Parameters:**

*No Parameters*

### Example:

```
GET http://magentohost/api/rest/customers/2
```

### Response Body:

```
<?xml version="1.0"?>
<magento_api>
 <entity_id>2</entity_id>
 <website_id>1</website_id>
 <email>test@example.com</email>
```

```

<group_id>1</group_id>
<created_at>2012-03-22 14:15:54</created_at>
<disable_auto_group_change>1</disable_auto_group_change>
<created_in>Default Store View</created_in>
<firstname>john</firstname>
<lastname>Doe</lastname>
<last_logged_in>2012-03-22 14:15:56</last_logged_in>
</magento_api>

```

#### response example: json

get <http://magentohost/api/rest/customers/141>

#### response body:

```
{
 "entity_id": "2",
 "website_id": "1",
 "email": "test@example.com",
 "group_id": "1",
 "created_at": "2012-03-22 14:15:54",
 "disable_auto_group_change": "1",
 "created_in": "English",
 "firstname": "john",
 "lastname": "Doe"
}
```

### HTTP Method: POST /customers/:customerId

**Description:** Not allowed.

### HTTP Method: PUT /customers/:customerId

**Description:** Allows you to update an existing customer.

**Notes:** The list of attributes that will be updated for customer is configured in the Magento Admin Panel. The Customer user type has access only to his/her own information.

**Authentication:** Admin, Customer

**Default Format:** XML

#### Parameters:

You must specify only those parameters which you want to update. Parameters that are not defined in the request body will preserve the previous values. The website\_id and created\_in attributes are not allowed for updating.

#### Example:

PUT http://magentohost/api/rest/customers/2

#### Request Body:

```

<?xml version="1.0"?>
<magento_api>
 <firstname>Earl</firstname>
 <lastname>Hickey</lastname>
 <email>customers@example.com</email>
 <group_id>1</group_id>
</magento_api>

```

## **HTTP Method: DELETE /customers/:customerId**

**Description:** Allows you to delete an existing customer.

**Notes:** Admin only can delete a customer.

**Authentication:** Admin

**Default Format:** XML

**Parameters:**

*No Parameters*

**Example:**

```
DELETE http://magentohost/api/rest/customers/2
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Response Formats

[Edit this page on GitHub](#) 

## Introduction

## Authentication

## Common HTTP Status Codes

## HTTP Methods

## GET Filters

## Permission Settings

## Resources

### Response Formats

## Testing REST Resources

## REST API Response Formats

### XML Format

### JSON Format

### Response Structure

### JSON Responses

If you make a Magento API call, you are guaranteed to receive some kind of a response. If you make a successful call, you will receive an HTTP response with a 200 OK status.

# REST API Response Formats

You can view the response data from any Magento API call in one of the following two formats:

- XML
- JSON

The format of returned data is defined in the request header. The format you choose depends on what you are familiar with most or tools available to you.

## XML Format

The XML response format is a simple XML block.

To set the response format to XML, add the Accept request header with the text/xml value.

A successful call will return the following response (example of retrieving information about stock items):

```
<?xml version="1.0"?>
<magento_api>
 <data_item>
 <item_id>1</item_id>
 <product_id>1</product_id>
 <stock_id>1</stock_id>
 <qty>99.0000</qty>
```

```

<low_stock_date></low_stock_date>
</data_item>
<data_item>
 <item_id>2</item_id>
 <product_id>2</product_id>
 <stock_id>1</stock_id>
 <qty>100.0000</qty>
 <low_stock_date></low_stock_date>
</data_item>
</magento_api>
```

If an error occurs, the call may return the following response:

```

<?xml version="1.0"?>
<magento_api>
 <messages>
 <error>
 <data_item>
 <code>404</code>
 <message>Resource not found.</message>
 </data_item>
 </error>
 </messages>
</magento_api>
```

## JSON Format

JSON (JavaScript Object Notation) is a lightweight data-interchange format.

To set the response format to JSON, add the Accept request header with the application/json value.

## Response Structure

The JSON objects represent a direct mapping of the XML block from the XML response format.

A simple XML error

```

<messages>
 <error>
 <data_item>
 <code>404</code>
 <message>Resource not found.</message>
 </data_item>
 </error>
</messages>
```

will be transformed to

```
{"messages":{"error":[{"code":404,"message":"Resource not found."}]} }
```

## JSON Responses

A successful API call to the Stock Items resource will return the following XML response:

```

<?xml version="1.0"?>
<magento_api>
 <data_item>
 <item_id>1</item_id>
 <product_id>1</product_id>
```

```
<stock_id>1</stock_id>
<qty>99.0000</qty>
<low_stock_date></low_stock_date>
</data_item>
<data_item>
<item_id>2</item_id>
<product_id>2</product_id>
<stock_id>1</stock_id>
<qty>100.0000</qty>
<low_stock_date></low_stock_date>
</data_item>
</magento_api>
```

The JSON equivalent will be as follows:

```
[{"item_id": "1", "product_id": "1", "stock_id": "1", "qty": "99.0000", "low_stock_date": null},
 {"item_id": "2", "product_id": "2", "stock_id": "1", "qty": "100.0000", "low_stock_date": null}]
```

The list of HTTP status codes that are returned in the API response is described in the [Common HTTP Status Codes](#) part of the documentation. There, you can find the list of codes themselves together with their description.

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Testing REST Resources

[Edit this page on GitHub](#) 

### Introduction

### Authentication

### Common HTTP Status Codes

### HTTP Methods

### GET Filters

### Permission Settings

### Resources

### Response Formats

### Testing REST Resources

Three steps are required for utilizing REST API resources:

- [Authenticate the user](#) (receive the access token for further steps);
- Configure the [permissions for operations](#) and [attributes](#) for the type of the user;
- Make an API call.

The following headers are required for the call:

- Authorization
- Version
- Accept
- Content-type

The following parameters must be provided in the Authorization header for the call:

- oauth\_signature\_method
- oauth\_version
- oauth\_nonce
- oauth\_timestamp
- oauth\_consumer\_key
- oauth\_token
- oauth\_signature

Testing REST resources with the [REST Client](#) plugin for the Mozilla Firefox browser.

1. Open the REST Client.
2. From the Authentication drop-down, select **OAuth**.

**OAuth**

|                            |                                                  |
|----------------------------|--------------------------------------------------|
| Signature for this request | OAuth setting                                    |
| Consumer key               | IPvnpgox1rS9xTy8p0DRtudP                         |
| Consumer secret            | IPvnpgox1rS9xTy8p0DRtudPePWNqtSjfkSqgFg          |
| Access token               | 1405721-iCeQLXdqTIEuYx2ipSipa1KkPhnbxiN9KJporshQ |
| Access token secret        | LgrNoNX4JNT3v03b2j6hPqQf6gHntUX5ui0CTLZs         |

Remember the setting      **Insert** ▾      **Close**

3. In the OAuth window, on the Signature for the request tab, fill in the following fields:

- **Consumer key:** Enter the **Key** value provided when you created the consumer in Magento Admin Panel.
- **Consumer secret:** Enter the **Secret** value provided when you created the consumer in Magento Admin Panel.
- **Access token:** Enter the oauth\_token value received when you authenticated the application.
- **Access token secret:** Enter the oauth\_token\_secret value received when you authenticated the application.

4. On the OAuth setting tab, define the following options:

**OAuth**

|                            |                        |
|----------------------------|------------------------|
| Signature for this request | OAuth setting          |
| Signature Methods          | HMAC-SHA1              |
| oAuth Version              | 1.0                    |
| Realm                      | http://sp.example.com/ |
| oAuth Nonce                |                        |
| oAuth Timestamp            |                        |

**Save**      **Close**

- **Signature Methods:** From the drop-down list, select which method will be used for signatures (HMAC-SHA1 or PLAINTEXT).
- **oAuth Version:** From the drop-down list, select the **1.0** option (REST API supports OAuth 1.0a).
- Leave the **Realm**, **oAuth Nonce**, and **oAuth Timestamp** values set by default.

5. Click **Save** and wait for the confirmation dialog to close.

6. Return to the Signature for the request tab and select **Insert > Insert as header**.

**OAuth**

|                                            |                                          |
|--------------------------------------------|------------------------------------------|
| <a href="#">Signature for this request</a> | <a href="#">OAuth setting</a>            |
| Consumer key                               | lsdijsldjwei949024jlsjj                  |
| Consumer secret                            | ldfjaldsfou438olavahalrgoa8fa0audsofhofh |
| Access token                               | askdufh084uoasd                          |
| Access token secret                        | akdua9s8aurfa98ufd                       |

Remember the setting [Insert as header](#) [Insert](#) [Close](#)

An authorization header is created on the main page of REST Client.

**[+] Request**

|                                                        |                                                                  |                      |
|--------------------------------------------------------|------------------------------------------------------------------|----------------------|
| Method: <a href="#">GET</a>                            | URL: <a href="http://www.example.com">http://www.example.com</a> | <a href="#">SEND</a> |
| <b>Headers</b>                                         |                                                                  |                      |
| Authorization: OAuth realm="*", o... <a href="#">X</a> |                                                                  |                      |
| <b>Body</b>                                            |                                                                  |                      |
| Request Body                                           |                                                                  |                      |

**NOTE:** Click the header with authorization data and click **Auto refresh** in the opened pop-up in order to generate new values for oauth\_nonce, oauth\_timestamp, and oauth\_signature at each request.

**OAuth**

```
OAuth oauth_version="1.0", oauth_signature_method="HMAC-SHA1",
oauth_nonce="ThVGHwkmkslyx83", oauth_timestamp="1342712491",
oauth_consumer_key="2myjt5x81yz4hzu057l0iqt8beh8sl",
oauth_token="q3tszqraj3td4ajwyctqw37bcw9w5g70",
oauth_signature="G8bCuGNKg4bi7jHG%2BkB6CnAYhYk%3D"
```

[Auto refresh](#) [Refresh](#) [Close](#)

7. From the **Headers** drop-down, select **Custom Header**.

The screenshot shows the RESTClient application interface. At the top, there are navigation menus: File, Authentication, Headers, View, Favorite Requests, Setting, and RESTClient logo. Below the header, there's a toolbar with Method (set to GET), URL (http://www.example.com), a star icon, and a red SEND button. A context menu is open over the URL field, with 'Custom Header' highlighted. The main workspace contains sections for Headers (with a single entry: Authorization: OAuth oauth\_versi...), Body (Request Body), and a footer with Home, Github, Issues, and Donate links, along with a Back to top button.

8. In the **Request Header** window, enter "Content-Type" in the **Name** field and "text/xml" in the **Value** field (if you want to use the XML data format). To use the JSON request data format, enter application/json instead of the text/xml value.
9. Click **Okay**.



#### Example: Retrieving the List of Products

1. From the **Method** drop-down list, select the **GET** option.
2. In the **URL** field, enter the following URL: http://magentohost/api/rest/products. You can limit the number of products returned in the response. To set the limit to 4, enter the following URL: http://magentohost/api/rest/products?limit=4
3. Click **Send**. Information about all products will be displayed in the response body. Example is as follows:

**[+] Response**

- [Response Headers](#)
- [Response Body \(Raw\)](#)
- [Response Body \(Highlight\)](#) **Response Body (Preview)**

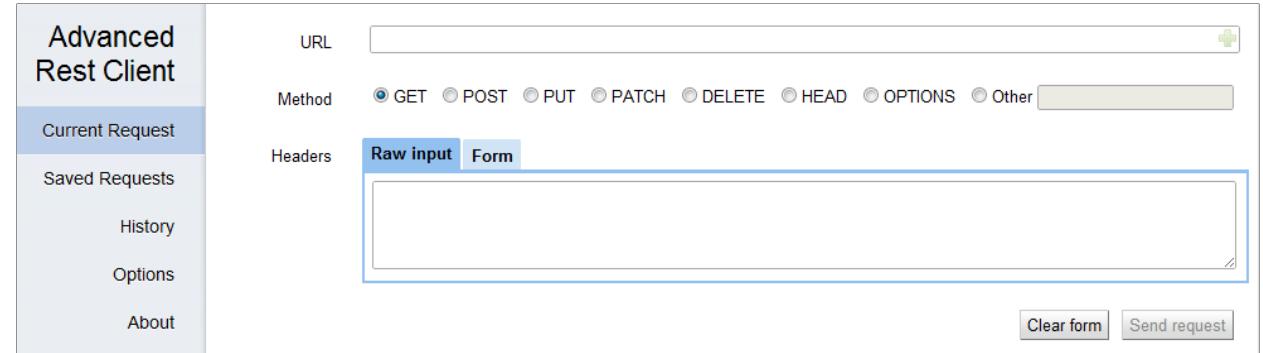
```

- <magento_api>
- <data_item>
 <entity_id>1</entity_id>
 <type_id>simple</type_id>
 <sku>n2610</sku>
- <description>
 The Nokia 2610 is an easy to use device that combines multiple messaging options including email, instant messaging, and more. You can even download MP3 ringtones, graphics, and games straight to the phone, or surf the Internet with Cingular's MEdia Net service. It's the perfect complement to Cingular service for those even remotely interested in mobile Web capabilities in an affordable handset. Design Compact and stylish, the 2610 features a candybar design sporting a bright 128 x 128 pixel display capable of displaying over 65,000 colors. Most of the phone's features and on-screen menus are controlled by a center toggle on the control pad. A standard hands-free headphone jack is provided, as are volume control keys, and there's even a "Go-To" button that can be assigned by the user for quick access to favorite applications. Lastly, the included speakerphone allows you to talk handsfree, and because the phone sports an internal antenna, there's nothing to snag or break off.
</description>
- <short_description>
 The words "entry level" no longer mean "low-end," especially when it comes to the Nokia 2610. Offering advanced media and calling features without breaking the bank
</short_description>
<meta_keyword/>
<name>Nokia 2610 Phone</name>
<meta_title/>
<meta_description/>
<final_price_with_tax>149.99</final_price_with_tax>
</data_item>
</magento_api>

```

Testing REST resources with the [Advanced REST Client](#) for Google Chrome browser.

1. Open the Advanced REST Client Application.



2. In the **Headers** box, select the **Form** tab.
3. In the first field, start typing *authorization*. An **Authorization** popup appears. Click it.

The screenshot shows the 'Advanced Rest Client' application. On the left is a sidebar with links: 'Current Request' (selected), 'Saved Requests', 'History', 'Options', and 'About'. The main area has tabs for 'URL' and 'Method' (set to GET). Under 'Headers', there are two tabs: 'Raw input' (selected) and 'Form'. In 'Raw input', there is a text input field containing 'auth' and a dropdown menu labeled 'Authorization'. A red box highlights this 'Authorization' section. At the bottom right are buttons for 'Clear form' and 'Send request'.

4. When you click the fields next to the Authorization header, the **Construct** link appears. Click it to configure OAuth authentication.

5. The Authorization window opens. Select the OAuth tab.

The screenshot shows the 'Authorization' dialog box. At the top are tabs for 'Basic' and 'OAuth' (selected). Below are fields for 'Realm' (empty), 'HTTP Method: GET', and 'Type' (radio buttons for 'Request Token', 'Request Access Token', and 'Signed Request', with 'Request Token' selected). Under 'signature method', 'HMAC-SHA1' is selected. There are fields for 'Consumer key' (text input 'Enter your key') and 'Consumer secret' (text input 'Enter your secret'). Below these are 'Timestamp' (text input '1341919961'), 'Nonce' (text input '5XPWECLYYKCZM7N2' with a 'Generate' link), and 'Version' (dropdown menu set to '1.0'). At the bottom are 'OK' and 'Cancel' buttons.

6. In the **Type** group of options, select the **Signed Request** option.

7. In the **signature method** group of options, select which method will be used for signatures (HMAC-SHA1 or PLAINTEXT).

8. Fill in the following data:

**Authorization**

**Basic**   **OAuth**

Realm:  
HTTP Method: GET

Type  
 Request Token    Request Access Token    Signed Request

signature method  
 HMAC-SHA1    PLAINTEXT

Consumer key: 9yx7go510gy4www05ltqx92k8jqopcb   Consumer secret: 5v008m5qwrao8nu8h04bov3x13h2oy4w

Access Token: q3tszqraj3td4ajwyctqw37bcw9w5g70   AccessToken Secret: d8j3wjuhgor038y3zgb3ibkt49g8nxz

Timestamp: 1341919961

Nonce: H8O1EULLYFV30NM7   [Generate](#)

Version: 1

**OK**   **Cancel**

- **Consumer key:** Enter the **Key** value provided when you created the consumer in Magento Admin Panel.
- **Consumer secret:** Enter the **Secret** value provided when you created the consumer in Magento Admin Panel.
- **Access Token:** Enter the oauth\_token value received when you authenticated the application.
- **Access Token Secret:** Enter the oauth\_token\_secret value received when you authenticated the application.

9. Click **OK**.

**NOTE:** Advanced REST Client does not save the **Consumer secret** and **Access Token Secret** values. You need to enter these values each time you make a request.

10. In the **URL** field, enter the URL to which the API call will be performed and select the required HTTP method.

11. In the **Headers** table, click **Add row** and add the Accept - application/json or Accept - text/xml header depending on which format you prefer for the returned data.

12. Click **Send Request**.

#### Example: Retrieving the list of customers

1. In the **Method** group of options, select the **GET** option.
2. In the URL field, enter the following URL: <http://magentohost/api/rest/customers>.
3. Click **Send request**. Information about all customers will be displayed in the response body. Note that only Admin type of the user can retrieve the list of customers. Example is as follows:

| Body | Raw response                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | JSON |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|      | <pre>{     - "4": {         "entity_id": "4",         "website_id": "2",         "email": "shon@example.com",         "group_id": "1",         "created_at": "2012-04-03 11:21:15",         "disable_auto_group_change": "0",         "firstname": "Earl",         "lastname": "Hickey",         "created_in": "Admin"     },     - "5": {         "entity_id": "5",         "website_id": "2",         "email": "donald@example.com",         "group_id": "1",         "created_at": "2012-04-06 12:57:34",         "disable_auto_group_change": "0",         "prefix": null,         "firstname": "Donald",         "middlename": null,         "lastname": "McDonald",         "suffix": null,         "taxvat": null,         "created_in": "Admin"     } }</pre> |      |

#### Example: Creating a customer address

1. In the **Method** group of options, select the **POST** option.
2. In the URL field, enter the following URL: <http://magentohost/api/rest/customers/:id/addresses> where the ":id" value is the customer ID in the system.
3. In the **Body** table, on the **Raw input** tab, enter the data required for customer address creation.
4. Click **Send request**. If the address is created, the 200 OK HTTP status code will be returned. Example is as follows:

**Advanced Rest Client**

URL: http://magento/api/rest/customers/6/addresses

Method: POST

Headers:

|           |      |
|-----------|------|
| Raw input | Form |
|-----------|------|

|                                                                             |                   |
|-----------------------------------------------------------------------------|-------------------|
| Authorization: OAuth realm="http://magento/",oauth_consumer_key="9yx7go510" | Remove            |
| Content-Type: text/xml                                                      | Edit value Remove |
| Accept: text/xml                                                            | Remove            |
| Add row                                                                     |                   |

Body:

|           |      |      |
|-----------|------|------|
| Raw input | Form | File |
|-----------|------|------|

```
<?xml version="1.0"?>
<magento_api>
 <firstname>Johny</firstname>
 <lastname>Doe</lastname>
```

Content-type: application/x-www-form-urlencoded

Content-Type header will overwrite this value.

Clear form Send request

Status code: 200 OK

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Introduction to the Magento 1.x SOAP API

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Introduction

### Supported Types

SOAP

XML-RPC

### API Methods

Global API Faults

SOAP API Version v2

## Introduction

The Magento SOAP v1 API provides you with the ability to manage your eCommerce stores by providing calls for working with resources such as customers, categories, products, and sales orders. It also allows you to manage shopping carts and inventory.

A SOAP v2 API version has been available since Magento 1.3, and a WS-I compliant version has been available since Magento 1.6.

## Supported Types

The Magento API supports [SOAP](#) and [XML-RPC](#), where SOAP is the default protocol.

## SOAP

To connect to Magento SOAP web services, load the [WSDL](#) into your SOAP client from either of these URLs:

`http://magentohost/api/?wsdl`

`http://magentohost/api/soap/?wsdl`

where magentohost is the domain for your Magento host.

As of v1.3, you may also use the following URL to access the Magento API v2, which has been added to improve compatibility with Java and .NET:

```
http://magentohost/api/v2_soap?wsdl=1
```

The following PHP example shows how to make SOAP calls to the Magento API v1:

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'somestuff.method');
$result = $client->call($session, 'somestuff.method', 'arg1');
$result = $client->call($session, 'somestuff.method', array('arg1', 'arg2', 'arg3'));
$result = $client->multiCall($session, array(
 array('somestuff.method'),
 array('somestuff.method', 'arg1'),
 array('somestuff.method', array('arg1', 'arg2'))
));

// If you don't need the session anymore
$client->endSession($session);
```

## XML-RPC

To use XML-RPC, load the following URL into your XML-RPC client:

```
http://magentohost/api/xmlrpc/
```

where magentohost is the domain for your Magento host.

The following PHP example shows how to make XML-RPC calls:

```
$client = new Zend_XmlRpc_Client('http://magentohost/api/xmlrpc/');

// If somestuff requires API authentication,
// we should get session token
$session = $client->call('login', array('apiUser', 'apiKey'));

$client->call('call', array($session, 'somestuff.method', array('arg1', 'arg2',
 'arg3')));
$client->call('call', array($session, 'somestuff.method', 'arg1'));
$client->call('call', array($session, 'somestuff.method'));
$client->call('multiCall', array($session,
 array(
 array('somestuff.method', 'arg1'),
 array('somestuff.method', array('arg1', 'arg2')),
 array('somestuff.method')
)
));

// If you don't need the session anymore
$client->call('endSession', array($session));
```

The XML-RPC only supports the version 1 of the Magento API.

## API Methods

The following table contains the API methods that can be called from your SOAP or XML-RPC client on the Magento v1 API.

| Method                                           | Description                                                                                                 | Return Value |
|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------|--------------|
| startSession()                                   | Start the API session and return session ID.                                                                | string       |
| endSession(sessionId)                            | End the API session.                                                                                        | boolean      |
| login(apiUser, apiKey)                           | Start the API session, return the session ID, and authorize the API user.                                   | string       |
| call(sessionId, resourcePath, array arguments)   | Call the API resource that is allowed in the current session. See Note below.                               | mixed        |
| multiCall(sessionId, array calls, array options) | Call the API resource's methods that are allowed for current session. See Notes below.                      | array        |
| resources(sessionId)                             | Return a list of available API resources and methods allowed for the current session.                       | array        |
| globalFaults(sessionId)                          | Return a list of fault messages and their codes that do not depend on any resource.                         | array        |
| resourceFaults(sessionId, resourceName)          | Return a list of the specified resource fault messages, if this resource is allowed in the current session. | array        |

**Note:** For **call** and **multiCall**, if no session is specified, you can call only resources that are not protected by ACL.

**Note:** For **multiCall**, if the "break" option is specified, multiCall breaks on first error.

The Magento SOAP API v2 does not support the call() and multiCall() methods, and instead provides a separate method for each API resource.

## Global API Faults

The following table contains fault codes that apply to all SOAP/XML-RPC API calls.

| Fault Code | Fault Message                               |
|------------|---------------------------------------------|
| 0          | Unknown Error                               |
| 1          | Internal Error. Please see log for details. |
| 2          | Access denied.                              |
| 3          | Invalid API path.                           |
| 4          | Resource path is not callable.              |

## SOAP API Version v2

Since Magento 1.3, version v2 of the SOAP API has also been available. The main difference between v1 and v2 is that instead of using methods call and multiCall, it has separate methods for each action.

For example, consider the following PHP code using SOAP v1.

```
$params = array(array(
 'status'=>array('eq'=>'pending'),
 'customer_is_guest'=>array('eq'=>'1'))
));
$result = $client->call($sessionId, 'sales_order.list', $params);
```

With SOAP v2, the following code would be equivalent.

```
$params = array('filter' => array(
 array('key' => 'status', 'value' => 'pending'),
 array('key' => 'customer_is_guest', 'value' => '1')
));
$result = $client->salesOrderList($sessionId, $params);
```

Note that the WSDL for SOAP v1 and SOAP v2 are different. Note that in SOAP v1, customizing the API did not involve changing the WSDL. In SOAP v2, changes to the WSDL are required.

You can configure the SOAP v2 API to be [WS-I](#) compliant in the system configuration menu. To do this, set **Services > Magento Core API > WS-I Compliance** to **Yes**.

Note that the WSDL for the SOAP v2 API is different when in WS-I compliant mode.

Using the WS-I compliant SOAP v2 API WSDL, it is easy to automatically generate client classes for Java, .NET, and other languages using standard libraries.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▾

[Catalog Category Attributes](#) ▾

[Catalog Product](#) ▾

[Catalog Product Attribute](#) ▾

[Catalog Product Attribute](#)

[Media](#) ▾

[Catalog Product Attribute Set](#) ▾

[Catalog Product Custom](#)

[Option](#) ▾

[Catalog Product Custom Option](#)

[Value](#) ▾

[Catalog Product Downloadable](#)

[Link](#) ▾

[Catalog Product Link](#) ▾

[Catalog Product Tag](#) ▾

[Catalog Product Tier Price](#) ▾

[Catalog Product Types](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Category

Allows you to manage categories and how products are assigned to categories.

**Resource Name:** catalog\_category

#### Aliases:

- category

#### Methods:

- [catalog\\_category.currentStore](#) - Set/Get the current store view
- [catalog\\_category.tree](#) - Retrieve the hierarchical category tree
- [catalog\\_category.level](#) - Retrieve one level of categories by a website, store view, or parent category
- [catalog\\_category.info](#) - Retrieve the category data
- [catalog\\_category.create](#) - Create a new category
- [catalog\\_category.update](#) - Update a category
- [catalog\\_category.move](#) - Move a category in its tree
- [catalog\\_category.delete](#) - Delete a category
- [catalog\\_category.assignedProducts](#) - Retrieve a list of products assigned to a category
- [catalog\\_category.assignProduct](#) - Assign product to a category
- [catalog\\_category.updateProduct](#) - Update an assigned product
- [catalog\\_category.removeProduct](#) - Remove a product assignment

### Category Attributes

Allows you to retrieve the list of category attributes and options.

**Resource Name:** catalog\_category\_attribute

**Aliases:**

- category\_attribute

**Methods:**

- [catalog\\_category\\_attribute.currentStore](#) - Set/Get the current store view
- [catalog\\_category\\_attribute.list](#) - Retrieve the category attributes
- [catalog\\_category\\_attribute.options](#) - Retrieve the attribute options

## Product

Allows you to manage products.

**Resource Name:** catalog\_product

**Aliases:**

- product

**Methods:**

- [catalog\\_product.currentStore](#) - Set/Get the current store view
- [catalog\\_product.list](#) - Retrieve the list of products using filters
- [catalog\\_product.info](#) - Retrieve information about the required product
- [catalog\\_product.create](#) - Create a new product
- [catalog\\_product.update](#) - Update a required product
- [catalog\\_product.setSpecialPrice](#) - Set special price for a product
- [catalog\\_product.getSpecialPrice](#) - Get special price for a product
- [catalog\\_product.delete](#) - Delete a required product
- [catalog\\_product.listOfAdditionalAttributes](#) - Get the list of additional attributes

## Product Attributes

Allows you to retrieve product attributes and options.

**Resource Name:** catalog\_product\_attribute

**Aliases:**

- product\_attribute

**Methods:**

- [product\\_attribute.currentStore](#) - Set/Get the current store view
- [product\\_attribute.list](#) - Retrieve the attribute list
- [product\\_attribute.options](#) - Retrieve the attribute options
- [product\\_attribute.addOption](#) - Add a new option for attributes with selectable fields
- [product\\_attribute.create](#) - Create a new attribute
- [product\\_attribute.info](#) - Get full information about an attribute with the list of options
- [product\\_attribute.remove](#) - Remove the required attribute
- [product\\_attribute.removeOption](#) - Remove an option for attributes with selectable fields
- [product\\_attribute.types](#) - Get the list of possible attribute types
- [product\\_attribute.update](#) - Update the required attribute

## Product Attribute Sets

Allows you to retrieve product attribute sets.

**Resource Name:** catalog\_product\_attribute\_set

**Aliases:**

- [product\\_attribute\\_set](#)

**Methods:**

- [product\\_attribute\\_set.list](#) - Retrieve the list of product attribute sets
- [product\\_attribute\\_set.attributeAdd](#) - Add an attribute to the attribute set
- [product\\_attribute\\_set.attributeRemove](#) - Remove an attribute from an attribute set
- [product\\_attribute\\_set.create](#) - Create a new attribute set
- [product\\_attribute\\_set.groupAdd](#) - Add a new group for attributes in the attribute set
- [product\\_attribute\\_set.groupRemove](#) - Remove a group of attributes from an attribute set
- [product\\_attribute\\_set.groupRename](#) - Rename a group of attributes in an attribute set
- [product\\_attribute\\_set.remove](#) - Remove an attribute set

## Product Types

Allows you to retrieve product types.

**Resource Name:** catalog\_product\_type

**Aliases:**

- [product\\_type](#)

**Methods:**

- [catalog\\_product\\_type.list](#) - Retrieve the list of product types

## Product Images

Allows you to manage product images.

**Resource Name:** catalog\_product\_attribute\_media

**Aliases:**

- [product\\_attribute\\_media](#)
- [product\\_media](#)

**Methods:**

- [catalog\\_product\\_attribute\\_media.currentStore](#) - Set/Get the current store view
- [catalog\\_product\\_attribute\\_media.list](#) - Retrieve the product images
- [catalog\\_product\\_attribute\\_media.info](#) - Retrieve the specified product image
- [catalog\\_product\\_attribute\\_media.types](#) - Retrieve product image types
- [catalog\\_product\\_attribute\\_media.create](#) - Upload a new image for a product
- [catalog\\_product\\_attribute\\_media.update](#) - Update an image for a product
- [catalog\\_product\\_attribute\\_media.remove](#) - Remove an image for a product

## Product Tier Price

Allows you to retrieve and update product tier prices.

**Resource Name:** catalog\_product\_attribute\_tier\_price

**Aliases:**

- product\_attribute\_tier\_price
- product\_tier\_price

**Methods:**

- [catalog\\_product\\_attribute\\_tier\\_price.info](#) - Retrieve information about product tier prices
- [catalog\\_product\\_attribute\\_tier\\_price.update](#) - Update the product tier prices

## Product Links

Allows you to manage links for products, including related, cross-sells, up-sells, and grouped.

**Resource Name:** catalog\_product\_link

**Aliases:**

- product\_link

**Methods:**

- [catalog\\_product\\_link.list](#) - Retrieve products linked to the specified product
- [catalog\\_product\\_link.assign](#) - Link a product to another product
- [catalog\\_product\\_link.update](#) - Update a product link
- [catalog\\_product\\_link.remove](#) - Remove a product link
- [catalog\\_product\\_link.types](#) - Retrieve product link types
- [catalog\\_product\\_link.attributes](#) - Retrieve product link type attributes

## Product Downloadable Link

Allows you to add, remove, and retrieve a link to a downloadable product.

**Resource Name:** product\_downloadable\_link

**Methods:**

- [product\\_downloadable\\_link.add](#) - Add a new link to the downloadable product
- [product\\_downloadable\\_link.list](#) - Get the list of links for a downloadable product
- [product\\_downloadable\\_link.remove](#) - Remove a link from a downloadable product

## Product Tag

Allows you to add, update, remove, and retrieve product tags.

**Resource Name:** catalog\_product\_tag

**Aliases:**

- product\_tag

**Methods:**

- [product\\_tag.add](#) - Retrieve the list of tags by the product ID
- [product\\_tag.info](#) - Retrieve information about a product tag
- [product\\_tag.add](#) - Add one or more tags to a product

- [product\\_tag.update](#) - Update an existing product tag
- [product\\_tag.remove](#) - Remove a product tag

## Product Custom Option

Allows you to manage product custom options, including adding, updating, removing, and retrieving information.

**Resource Name:** catalog\_product\_custom\_option

**Aliases:**

- [product\\_custom\\_option](#)

**Methods:**

- [product\\_custom\\_option.add](#) - Add a new custom option to a product
- [product\\_custom\\_option.update](#) - Update the product custom option
- [product\\_custom\\_option.types](#) - Get the list of available custom option types
- [product\\_custom\\_option.list](#) - Retrieve the list of product custom options
- [product\\_custom\\_option.info](#) - Get full information about the custom option in a product
- [product\\_custom\\_option.remove](#) - Remove the custom option

## Product Custom Option Value

Allows you to manage product custom option values, including adding, updating, removing, and retrieving information.

**Resource Name:** catalog\_product\_custom\_option\_value

**Aliases:**

- [product\\_custom\\_option\\_value](#)

**Methods:**

- [product\\_custom\\_option\\_value.add](#) - Add a new custom option value to a selectable custom option
- [product\\_custom\\_option\\_value.list](#) - Retrieve the list of product custom option values
- [product\\_custom\\_option\\_value.info](#) - Retrieve full information about the specified product custom option value
- [product\\_custom\\_option\\_value.update](#) - Update the custom option value
- [product\\_custom\\_option\\_value.remove](#) - Remove the custom option value

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Category

[Edit this page on GitHub](#) 

## Introduction

### Catalog

#### Catalog Category

`catalog_category.assignedPro...`  
`catalog_category.assignProduct`  
`catalog_category.create`  
`catalog_category.currentStore`  
`catalog_category.delete`  
`catalog_category.info`  
`catalog_category.level`  
`catalog_category.move`  
`catalog_category.removeProduct`  
`catalog_category.tree`  
`catalog_category.update`  
`catalog_category.updateProduct`

#### Catalog Category Attributes

#### Catalog Product

#### Catalog Product Attribute

#### Catalog Product Attribute

#### Media

#### Catalog Product Attribute Set

#### Catalog Product Custom

#### Option

#### Catalog Product Custom Option Value

#### Catalog Product Downloadable Link

#### Catalog Product Link

#### Catalog Product Tag

#### Catalog Product Tier Price

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Category

Allows you to manage categories and how products are assigned to categories.

**Resource Name:** catalog\_category

#### Aliases:

- category

#### Methods:

- `catalog_category.currentStore` - Set/Get the current store view
- `catalog_category.tree` - Retrieve the hierarchical category tree
- `catalog_category.level` - Retrieve one level of categories by a website, store view, or parent category
- `catalog_category.info` - Retrieve the category data
- `catalog_category.create` - Create a new category
- `catalog_category.update` - Update a category
- `catalog_category.move` - Move a category in its tree
- `catalog_category.delete` - Delete a category
- `catalog_category.assignedProducts` - Retrieve a list of products assigned to a category
- `catalog_category.assignProduct` - Assign product to a category
- `catalog_category.updateProduct` - Update an assigned product
- `catalog_category.removeProduct` - Remove a product assignment

### Faults

| Fault Code | Fault Message |
|------------|---------------|
|            |               |

**Catalog Product Types** ▾**Catalog Inventory** ▾**Checkout** ▾**Create Your Own API****Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

|     |                                                 |
|-----|-------------------------------------------------|
| 100 | Requested store view not found.                 |
| 101 | Requested website not found.                    |
| 102 | Category not exists.                            |
| 103 | Invalid data given. Details in error message.   |
| 104 | Category not moved. Details in error message.   |
| 105 | Category not deleted. Details in error message. |
| 106 | Requested product is not assigned to category.  |

**Example 1. Working with categories**

```

function getSomeRandomCategory(&$categories, $targetLevel, $currentLevel = 0) {
 if (count($categories)==0) {
 return false;
 }
 if ($targetLevel == $currentLevel) {
 return $categories[array_rand($categories)];
 } else {
 return getSomeRandomCategory($categories[array_rand($categories)]['children'],
$targetLevel + 1);
 }
}

$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$allCategories = $proxy->call($sessionId, 'category.tree'); // Get all categories.

// select random category from tree
while (($selectedCategory = getSomeRandomCategory($allCategories, 3)) === false) {}

// create new category
$newCategoryId = $proxy->call(
 $sessionId,
 'category.create',
 array(
 $selectedCategory['category_id'],
 array(
 'name'=>'Newopenerp',
 'is_active'=>1,
 'include_in_menu'=>2,
 'available_sort_by'=>'position',
 'default_sort_by'=>'position'
)
)
);

$newData = array('is_active'=>1);
// update created category on German store view
$proxy->call($sessionId, 'category.update', array($newCategoryId, $newData,
'german'));

$firstLevel = $proxy->call($sessionId, 'category.level', array(null, 'german',
$selectedCategory['category_id']));

var_dump($firstLevel);

// If you wish remove category, uncomment next line

```

```
//$proxy->call($sessionId, 'category.delete', $newCategoryId);
```

## Example 2. Working with assigned products

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$categoryId = 5; // Put here your category id
$storeId = 1; // You can add store level

$assignedProducts = $proxy->call($sessionId, 'category.assignedProducts',
array($categoryId, $storeId));
var_dump($assignedProducts); // Will output assigned products.

// Assign product
$proxy->call($sessionId, 'category.assignProduct', array($categoryId,
'someProductSku', 5));

// Update product assignment position
$proxy->call($sessionId, 'category.updateProduct', array($categoryId,
'someProductSku', 25));

// Remove product assignment
$proxy->call($sessionId, 'category.removeProduct', array($categoryId,
'someProductSku'));
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Assigned Products

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

### Catalog Category ^

- [catalog\\_category.assignedProducts](#)
- [catalog\\_category.assignProduct](#)
- [catalog\\_category.create](#)
- [catalog\\_category.currentStore](#)
- [catalog\\_category.delete](#)
- [catalog\\_category.info](#)
- [catalog\\_category.level](#)
- [catalog\\_category.move](#)
- [catalog\\_category.removeProduct](#)
- [catalog\\_category.tree](#)
- [catalog\\_category.update](#)
- [catalog\\_category.updateProduct](#)

### Catalog Category Attributes ^

### Catalog Product ^

#### Catalog Product Attribute ^

#### Catalog Product Attribute

### Media ^

#### Catalog Product Attribute Set ^

#### Catalog Product Custom

### Option ^

#### Catalog Product Custom Option

### Value ^

#### Catalog Product Downloadable

### Link ^

#### Catalog Product Link ^

#### Catalog Product Tag ^

#### Catalog Product Tier Price ^

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

Resource Name: catalog\_category

### Aliases:

- category

### Method:

- catalog\_category.assignedProducts (SOAP V1)
- catalogCategoryAssignedProducts (SOAP V2)

Retrieve the list of products assigned to a required category.

### Aliases:

- category.assignedProducts

### Arguments:

| Type   | Name       | Description                 |
|--------|------------|-----------------------------|
| string | sessionId  | Session ID                  |
| int    | categoryId | ID of the required category |

### Returns:

| Type  | Name   | Description                     |
|-------|--------|---------------------------------|
| array | result | Array of catalogAssignedProduct |

The **catalogAssignedProduct** content is as follows:

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

## Catalog Product Types ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

| Type   | Name       | Description                      |
|--------|------------|----------------------------------|
| int    | product_id | ID of the assigned product       |
| string | type       | Product type                     |
| int    | set        | Attribute set ID                 |
| string | sku        | Product SKU                      |
| int    | position   | Position of the assigned product |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category.assignedProducts', '4');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryAssignedProducts($sessionId, '4');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogCategoryAssignedProducts((object)array('sessionId' => $sessionId->result, 'categoryId' => '4'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 0 =>
 array
 'product_id' => string '1' (length=1)
 'type' => string 'simple' (length=6)
 'set' => string '4' (length=1)
 'sku' => string 'n2610' (length=5)
 'position' => string '1' (length=1)
 1 =>
```

```
array
 'product_id' => string '2' (length=1)
 'type' => string 'simple' (length=6)
 'set' => string '4' (length=1)
 'sku' => string 'b8100' (length=5)
 'position' => string '1' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Assign Product

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

### Catalog Category ^

catalog\_category.assignedPro...  
[catalog\\_category.assignProduct](#)  
catalog\_category.create  
catalog\_category.currentStore  
catalog\_category.delete  
catalog\_category.info  
catalog\_category.level  
catalog\_category.move  
catalog\_category.removeProduct  
catalog\_category.tree  
catalog\_category.update  
catalog\_category.updateProduct

### Catalog Category Attributes ^

### Catalog Product ^

#### Catalog Product Attribute ^

#### Catalog Product Attribute

### Media ^

#### Catalog Product Attribute Set ^

#### Catalog Product Custom

#### Option ^

#### Catalog Product Custom Option

#### Value ^

#### Catalog Product Downloadable

#### Link ^

#### Catalog Product Link ^

#### Catalog Product Tag ^

#### Catalog Product Tier Price ^

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

Resource Name: catalog\_category

### Aliases:

- category

### Method:

- catalog\_category.assignProduct (SOAP V1)
- catalogCategoryAssignProduct (SOAP V2)

Assign a product to the required category.

### Aliases:

- category.assignProduct

### Arguments:

| Type   | Name              | Description                                                               |
|--------|-------------------|---------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                |
| int    | categoryId        | ID of the category                                                        |
| string | product/productId | ID or SKU of the product to be assigned to the category                   |
| string | position          | Position of the assigned product in the category (optional)               |
| string | identifierType    | Defines whether the product ID or SKU is passed in the 'product' argument |

### Returns:

## Catalog Product Types ▾

| Type    | Description                                               |
|---------|-----------------------------------------------------------|
| boolean | True if the product is assigned to the specified category |

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category.assignProduct', array('categoryId'
=> '4', 'product' => '1'));
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if
necessary

$result = $proxy->catalogCategoryAssignProduct($sessionId, '4', '3');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogCategoryAssignProduct((object)array('sessionId' =>
$sessionId->result, 'categoryId' => '5', 'productId' => '1', 'position' => '5'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Create Category

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

### Catalog Category ^

catalog\_category.assignedPro...  
catalog\_category.assignProduct  
**catalog\_category.create**  
catalog\_category.currentStore  
catalog\_category.delete  
catalog\_category.info  
catalog\_category.level  
catalog\_category.move  
catalog\_category.removeProduct  
catalog\_category.tree  
catalog\_category.update  
catalog\_category.updateProduct

### Catalog Category Attributes ^

### Catalog Product ^

#### Catalog Product Attribute ^

#### Catalog Product Attribute

### Media ^

### Catalog Product Attribute Set ^

### Catalog Product Custom

#### Option ^

### Catalog Product Custom Option

#### Value ^

### Catalog Product Downloadable

#### Link ^

### Catalog Product Link ^

### Catalog Product Tag ^

### Catalog Product Tier Price ^

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Resource Name: catalog\_category

#### Aliases:

- category

#### Method:

- catalog\_category.create (SOAP V1)
- catalogCategoryCreate (SOAP V2)

Create a new category and return its ID.

#### Aliases:

- category.create

#### Arguments:

| Type   | Name         | Description                          |
|--------|--------------|--------------------------------------|
| string | sessionId    | Session ID                           |
| int    | parentId     | Parent category ID                   |
| array  | categoryData | Array of catalogCategoryEntityCreate |
| string | storeView    | Store view ID or code (optional)     |

#### Returns:

| Type | Name | Description |
|------|------|-------------|
|      |      |             |

**Catalog Product Types** ▾

|     |              |                            |
|-----|--------------|----------------------------|
| int | attribute_id | ID of the created category |
|-----|--------------|----------------------------|

**Catalog Inventory** ▾**Checkout** ▾**Create Your Own API****Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**The **categoryData** content is as follows:

| Type          | Name                       | Description                                                                                                                          |
|---------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| string        | name                       | Name of the created category                                                                                                         |
| int           | is_active                  | Defines whether the category will be visible in the frontend                                                                         |
| int           | position                   | Position of the created category (optional)                                                                                          |
| ArrayOfString | available_sort_by          | All available options by which products in the category can be sorted                                                                |
| string        | custom_design              | The custom design for the category (optional)                                                                                        |
| int           | custom_apply_to_products   | Apply the custom design to all products assigned to the category (optional)                                                          |
| string        | custom_design_from         | Date starting from which the custom design will be applied to the category (optional)                                                |
| string        | custom_design_to           | Date till which the custom design will be applied to the category (optional)                                                         |
| string        | custom_layout_update       | Custom layout update (optional)                                                                                                      |
| string        | default_sort_by            | The default option by which products in the category are sorted                                                                      |
| string        | description                | Category description (optional)                                                                                                      |
| string        | display_mode               | Content that will be displayed on the category view page (optional)                                                                  |
| int           | is_anchor                  | Defines whether the category will be anchored (optional)                                                                             |
| int           | landing_page               | Landing page (optional)                                                                                                              |
| string        | meta_description           | Category meta description (optional)                                                                                                 |
| string        | meta_keywords              | Category meta keywords (optional)                                                                                                    |
| string        | meta_title                 | Category meta title (optional)                                                                                                       |
| string        | page_layout                | Type of page layout that the category should use (optional)                                                                          |
| string        | url_key                    | A relative URL path which can be entered in place of the standard target path (optional)                                             |
| int           | include_in_menu            | Defines whether the category is visible on the top menu bar                                                                          |
| string        | filter_price_range         | Price range of each price level displayed in the layered navigation block (optional)                                                 |
| int           | custom_use_parent_settings | Defines whether the category will inherit custom design settings of the category to which it is assigned. 1 - Yes, 0 - No (optional) |

**Notes:** The **position** parameter is deprecated, the category will be positioned anyway in the end of the list and you can not set the position directly. You should use the `catalog_category.move` method instead. You cannot also assign a root category to the specified store.

**Examples****Request Example SOAP V1**

```

$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->call($session, 'catalog_category.create', array(2, array(
 'name' => 'Category name',
 'is_active' => 1,
 'position' => 1,
 //<!-- position parameter is deprecated, category anyway will be positioned in the
end of list
 //and you can not set position directly, use catalog_category.move instead -->
 'available_sort_by' => 'position',
 'custom_design' => null,
 'custom_apply_to_products' => null,
 'custom_design_from' => null,
 'custom_design_to' => null,
 'custom_layout_update' => null,
 'default_sort_by' => 'position',
 'description' => 'Category description',
 'display_mode' => null,
 'is_anchor' => 0,
 'landing_page' => null,
 'meta_description' => 'Category meta description',
 'meta_keywords' => 'Category meta keywords',
 'meta_title' => 'Category meta title',
 'page_layout' => 'two_columns_left',
 'url_key' => 'url-key',
 'include_in_menu' => 1,
));
var_dump ($result);
</pre>

```

## Request Example SOAP V2

```

$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->catalogCategoryCreate($session, 2, array(
 'name' => 'Category name 2',
 'is_active' => 1,
 'position' => 1,
 //<!-- position parameter is deprecated, category anyway will be positioned in the
end of list
 //and you can not set position directly, use catalog_category.move instead -->
 'available_sort_by' => array('position'),
 'custom_design' => null,
 'custom_apply_to_products' => null,
 'custom_design_from' => null,
 'custom_design_to' => null,
 'custom_layout_update' => null,
 'default_sort_by' => 'position',
 'description' => 'Category description',
 'display_mode' => null,
 'is_anchor' => 0,
 'landing_page' => null,
 'meta_description' => 'Category meta description',
 'meta_keywords' => 'Category meta keywords',
 'meta_title' => 'Category meta title',
 'page_layout' => 'two_columns_left',
 'url_key' => 'url-key',
</pre>

```

```
 'include_in_menu' => 1,
));

var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogCategoryCreate((object)array('sessionId' => $sessionId-
>result, 'parentId' => '5', 'categoryData' => ((object)array(
 'name' => 'category',
 'is_active' => '1',
 'position' => '1',
 'available_sort_by' => array('position'),
 'default_sort_by' => 'position',
 'description' => 'Category description',
 'is_anchor' => '1',
 'include_in_menu' => '1'
))));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Current Store

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

#### Catalog Category ^

catalog\_category.assignedPro...  
catalog\_category.assignProduct  
catalog\_category.create  
**catalog\_category.currentStore**  
catalog\_category.delete  
catalog\_category.info  
catalog\_category.level  
catalog\_category.move  
catalog\_category.removeProduct  
catalog\_category.tree  
catalog\_category.update  
catalog\_category.updateProduct

#### Catalog Category Attributes ^

#### Catalog Product ^

#### Catalog Product Attribute ^

#### Catalog Product Attribute

#### Media ^

#### Catalog Product Attribute Set ^

#### Catalog Product Custom

#### Option ^

#### Catalog Product Custom Option

#### Value ^

#### Catalog Product Downloadable

#### Link ^

#### Catalog Product Link ^

#### Catalog Product Tag ^

#### Catalog Product Tier Price ^

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

Resource Name: catalog\_category

#### Aliases:

- category

#### Method:

- catalog\_category.currentStore (SOAP V1)
- catalogCategoryCurrentStore (SOAP V2)

Allows you to set/get the current store view.

#### Aliases:

- category.currentStore

#### Arguments:

| Type   | Name      | Description           |
|--------|-----------|-----------------------|
| string | sessionId | Session ID            |
| string | storeView | Store view ID or code |

#### Returns:

| Type | Name      | Description   |
|------|-----------|---------------|
| int  | storeView | Store view ID |

## Examples

## Catalog Product Types ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

---

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'category.currentStore', '1');
var_dump ($result);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryCurrentStore($sessionId, '1');
var_dump($result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Category Delete

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

### Catalog Category ^

catalog\_category.assignedPro...  
catalog\_category.assignProduct  
catalog\_category.create  
catalog\_category.currentStore  
[catalog\\_category.delete](#)  
catalog\_category.info  
catalog\_category.level  
catalog\_category.move  
catalog\_category.removeProduct  
catalog\_category.tree  
catalog\_category.update  
catalog\_category.updateProduct

### Catalog Category Attributes ^

### Catalog Product ^

#### Catalog Product Attribute ^

#### Catalog Product Attribute

### Media ^

### Catalog Product Attribute Set ^

### Catalog Product Custom

### Option ^

### Catalog Product Custom Option

### Value ^

### Catalog Product Downloadable

### Link ^

### Catalog Product Link ^

### Catalog Product Tag ^

### Catalog Product Tier Price ^

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

Resource Name: catalog\_category

#### Aliases:

- category

#### Method:

- catalog\_category.delete (SOAP V1)
- catalogCategoryDelete (SOAP V2)

Allows you to delete the required category.

#### Aliases:

- category.delete

#### Arguments:

| Type   | Name       | Description                      |
|--------|------------|----------------------------------|
| string | sessionId  | Session ID                       |
| int    | categoryId | ID of the category to be deleted |

#### Returns:

| Type    | Description                     |
|---------|---------------------------------|
| boolean | True if the category is deleted |

## Examples

## Catalog Product Types ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category.delete', '7');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryDelete($sessionId, '7');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogCategoryDelete((object)array('sessionId' => $sessionId->result, 'categoryId' => '7'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Category Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog

#### Catalog Category

catalog\_category.assignedPro...  
catalog\_category.assignProduct  
catalog\_category.create  
catalog\_category.currentStore  
catalog\_category.delete  
[catalog\\_category.info](#)  
catalog\_category.level  
catalog\_category.move  
catalog\_category.removeProduct  
catalog\_category.tree  
catalog\_category.update  
catalog\_category.updateProduct

#### Catalog Category Attributes

#### Catalog Product

#### Catalog Product Attribute

#### Catalog Product Attribute

#### Media

#### Catalog Product Attribute Set

#### Catalog Product Custom

#### Option

#### Catalog Product Custom Option

#### Value

#### Catalog Product Downloadable

#### Link

#### Catalog Product Link

#### Catalog Product Tag

#### Catalog Product Tier Price

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

Resource Name: catalog\_category

#### Aliases:

- category

#### Method:

- catalog\_category.info (SOAP V1)
- catalogCategoryInfo (SOAP V2)

Allows you to retrieve information about the required category.

#### Aliases:

- category.info

#### Arguments:

| Type          | Name       | Description                      |
|---------------|------------|----------------------------------|
| string        | sessionId  | Session ID                       |
| int           | categoryId | Category ID                      |
| string        | storeView  | Store view ID or code (optional) |
| ArrayOfString | attributes | Array of attributes (optional)   |

#### Returns:

| Type | Name | Description |
|------|------|-------------|
|      |      |             |

**Catalog Product Types** ▾

|       |      |                              |
|-------|------|------------------------------|
| array | info | Array of catalogCategoryInfo |
|-------|------|------------------------------|

**Catalog Inventory** ▾

The **catalogCategoryInfo** content is as follows:

**Checkout** ▾**Create Your Own API****Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

| Type          | Name                     | Description                                                                              |
|---------------|--------------------------|------------------------------------------------------------------------------------------|
| string        | category_id              | Category ID                                                                              |
| int           | is_active                | Defines whether the category is active                                                   |
| string        | position                 | Category position                                                                        |
| string        | level                    | Category level                                                                           |
| string        | parent_id                | Parent category ID                                                                       |
| string        | all_children             | All child categories of the current category                                             |
| string        | children                 | Names of direct child categories                                                         |
| string        | created_at               | Date when the category was created                                                       |
| string        | updated_at               | Date when the category was updated                                                       |
| string        | name                     | Category name                                                                            |
| string        | url_key                  | A relative URL path which can be entered in place of the standard target path (optional) |
| string        | description              | Category description                                                                     |
| string        | meta_title               | Category meta title                                                                      |
| string        | meta_keywords            | Category meta keywords                                                                   |
| string        | meta_description         | Category meta description                                                                |
| string        | path                     | Path                                                                                     |
| string        | url_path                 | URL path                                                                                 |
| int           | children_count           | Number of child categories                                                               |
| string        | display_mode             | Content that will be displayed on the category view page (optional)                      |
| int           | is_anchor                | Defines whether the category is anchored                                                 |
| ArrayOfString | available_sort_by        | All available options by which products in the category can be sorted                    |
| string        | custom_design            | The custom design for the category (optional)                                            |
| string        | custom_apply_to_products | Apply the custom design to all products assigned to the category (optional)              |
| string        | custom_design_from       | Date starting from which the custom design will be applied to the category (optional)    |
| string        | custom_design_to         | Date till which the custom design will be applied to the category (optional)             |
| string        | page_layout              | Type of page layout that the category should use (optional)                              |
| string        | custom_layout_update     | Custom layout update (optional)                                                          |
| string        | default_sort_by          | The default option by which products in the category are sorted                          |

|        |                            |                                                                                                                           |
|--------|----------------------------|---------------------------------------------------------------------------------------------------------------------------|
| int    | landing_page               | Landing page (optional)                                                                                                   |
| int    | include_in_menu            | Defines whether the category is available on the Magento top menu bar                                                     |
| string | filter_price_range         | Price range of each price level displayed in the layered navigation block                                                 |
| int    | custom_use_parent_settings | Defines whether the category will inherit custom design settings of the category to which it is assigned. 1 - Yes, 0 - No |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category.info', '5');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryInfo($sessionId, '5');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogCategoryInfo((object)array('sessionId' => $sessionId->result,
'categoryId' => '5'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 'category_id' => string '5' (length=1)
 'is_active' => string '1' (length=1)
 'position' => string '1' (length=1)
 'level' => string '2' (length=1)
 'parent_id' => int 3
 'increment_id' => null
 'created_at' => string '2012-03-29 12:30:51' (length=19)
 'updated_at' => string '2012-03-29 14:25:08' (length=19)
 'name' => string 'Mobile Phones' (length=13)
 'url_key' => string 'mobile-phones' (length=13)
 'thumbnail' => null
```

```
'description' => string 'Category for cell phones' (length=24)
'image' => null
'meta_title' => string 'Cell Phones' (length=11)
'meta_keywords' => string 'cell, phone' (length=11)
'meta_description' => null
'include_in_menu' => string '1' (length=1)
'path' => string '1/3/4' (length=5)
'all_children' => string '4' (length=1)
'path_in_store' => null
'children' => string '' (length=0)
'url_path' => string 'mobile-phones.html' (length=18)
'children_count' => string '0' (length=1)
'display_mode' => string 'PRODUCTS' (length=8)
'landing_page' => null
'is_anchor' => string '1' (length=1)
'available_sort_by' => null
'default_sort_by' => null
'filter_price_range' => null
'custom_use_parent_settings' => string '1' (length=1)
'custom_apply_to_products' => null
'custom_design' => null
'custom_design_from' => null
'custom_design_to' => null
'page_layout' => null
'custom_layout_update' => null
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Category Level

[Edit this page on GitHub](#) 

### Introduction

### Catalog

#### Catalog Category

- catalog\_category.assignedPro...
- catalog\_category.assignProduct
- catalog\_category.create
- catalog\_category.currentStore
- catalog\_category.delete
- catalog\_category.info
- catalog\_category.level**
- catalog\_category.move
- catalog\_category.removeProduct
- catalog\_category.tree
- catalog\_category.update
- catalog\_category.updateProduct

#### Catalog Category Attributes

#### Catalog Product

#### Catalog Product Attribute

#### Catalog Product Attribute

#### Media

#### Catalog Product Attribute Set

#### Catalog Product Custom

#### Option

#### Catalog Product Custom Option

#### Value

#### Catalog Product Downloadable

#### Link

#### Catalog Product Link

#### Catalog Product Tag

#### Catalog Product Tier Price

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

Resource Name: catalog\_category

#### Aliases:

- category

#### Method:

- catalog\_category.level (SOAP V1)
- catalogCategoryLevel (SOAP V2)

Allows you to retrieve one level of categories by a website, a store view, or a parent category.

#### Aliases:

- category.level

#### Arguments:

| Type   | Name           | Description                      |
|--------|----------------|----------------------------------|
| string | sessionId      | Session ID                       |
| string | website        | Website ID or code (optional)    |
| string | storeView      | Store view ID or code (optional) |
| string | parentCategory | Parent category ID (optional)    |

#### Returns:

| Type | Name | Description |
|------|------|-------------|
|      |      |             |

**Catalog Product Types** ▾

|       |      |                                            |
|-------|------|--------------------------------------------|
| array | tree | Array of CatalogCategoryEntitiesNoChildren |
|-------|------|--------------------------------------------|

**Catalog Inventory** ▾**Checkout** ▾**Create Your Own API****Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

The **CatalogCategoryEntityNoChildren** content is as follows:

| Type   | Name        | Description                            |
|--------|-------------|----------------------------------------|
| int    | category_id | Category ID                            |
| int    | parent_id   | Parent category ID                     |
| string | name        | Category name                          |
| int    | is_active   | Defines whether the category is active |
| int    | position    | Category position                      |
| int    | level       | Category level                         |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category.level');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryLevel($sessionId);
var_dump($result);
```

### Response Example SOAP V1

```
array
0 =>
array
'category_id' => string '2' (length=1)
'parent_id' => int 1
'name' => string 'Default Category' (length=16)
'is_active' => string '1' (length=1)
'position' => string '1' (length=1)
'level' => string '1' (length=1)
1 =>
array
'category_id' => string '3' (length=1)
'parent_id' => int 1
'name' => string 'root_category' (length=13)
'is_active' => string '1' (length=1)
```

```
'position' => string '2' (length=1)
'level' => string '1' (length=1)
```

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Category Move

[Edit this page on GitHub](#) 

### Introduction

### Catalog

#### Catalog Category

catalog\_category.assignedPro...  
catalog\_category.assignProduct  
catalog\_category.create  
catalog\_category.currentStore  
catalog\_category.delete  
catalog\_category.info  
catalog\_category.level  
[catalog\\_category.move](#)  
catalog\_category.removeProduct  
catalog\_category.tree  
catalog\_category.update  
catalog\_category.updateProduct

#### Catalog Category Attributes

#### Catalog Product

#### Catalog Product Attribute

#### Catalog Product Attribute

#### Media

#### Catalog Product Attribute Set

#### Catalog Product Custom

#### Option

#### Catalog Product Custom Option

#### Value

#### Catalog Product Downloadable

#### Link

#### Catalog Product Link

#### Catalog Product Tag

#### Catalog Product Tier Price

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

Resource Name: catalog\_category

#### Aliases:

- category

#### Method:

- catalog\_category.move (SOAP V1)
- catalogCategoryMove (SOAP V2)

Allows you to move the required category in the category tree.

#### Aliases:

- category.move

#### Arguments:

| Type   | Name       | Description                                                                                 |
|--------|------------|---------------------------------------------------------------------------------------------|
| string | sessionId  | Session ID                                                                                  |
| int    | categoryId | ID of the category to be moved                                                              |
| int    | parentId   | ID of the new parent category                                                               |
| string | afterId    | ID of the category after which the required category will be moved (optional for V1 and V2) |

#### Returns:

| Type | Name | Description |
|------|------|-------------|
|      |      |             |

## Catalog Product Types ▾

|         |    |                               |
|---------|----|-------------------------------|
| boolean | id | True if the category is moved |
|---------|----|-------------------------------|

## Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category.move', array('categoryId' => '4',
'parentId' => '3'));
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryMove($sessionId, '4', '3');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogCategoryMove((object)array('sessionId' => $sessionId->result,
'categoryId' => '19', 'parentId' => '8', 'afterId' => '4'));
var_dump($result->result);
```

**Note:** Please make sure that you are not moving the category to any of its own children. There are no extra checks to prevent doing it through API, and you won't be able to fix this from the admin interface later.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Remove Product

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

#### Catalog Category ^

catalog\_category.assignedPro...  
catalog\_category.assignProduct  
catalog\_category.create  
catalog\_category.currentStore  
catalog\_category.delete  
catalog\_category.info  
catalog\_category.level  
catalog\_category.move  
**catalog\_category.removeProduct**  
catalog\_category.tree  
catalog\_category.update  
catalog\_category.updateProduct

#### Catalog Category Attributes ^

#### Catalog Product ^

#### Catalog Product Attribute ^

#### Catalog Product Attribute

#### Media ^

#### Catalog Product Attribute Set ^

#### Catalog Product Custom

#### Option ^

#### Catalog Product Custom Option

#### Value ^

#### Catalog Product Downloadable

#### Link ^

#### Catalog Product Link ^

#### Catalog Product Tag ^

#### Catalog Product Tier Price ^

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

Resource Name: catalog\_category

### Aliases:

- category

### Method:

- catalog\_category.removeProduct (SOAP V1)
- catalogCategoryRemoveProduct (SOAP V2)

Allows you to remove the product assignment from the category.

### Aliases:

- category.removeProduct

### Arguments:

| Type   | Name           | Description                                                                |
|--------|----------------|----------------------------------------------------------------------------|
| string | sessionId      | Session ID                                                                 |
| int    | categoryId     | Category ID                                                                |
| string | productId      | ID or SKU of the product to be removed from the category                   |
| string | identifierType | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

| Type | Description |
|------|-------------|
|      |             |

## Catalog Product Types ▾

|         |                                                  |
|---------|--------------------------------------------------|
| boolean | True if the product is removed from the category |
|---------|--------------------------------------------------|

## Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category.removeProduct', array('categoryId'
=> '4', 'productId' => '3'));
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if
necessary

$result = $proxy->catalogCategoryRemoveProduct($sessionId, '4', '3');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogCategoryRemoveProduct((object)array('sessionId' =>
$sessionId->result, 'categoryId' => '4', 'productId' => '3'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Category Tree

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

### Catalog Category ^

catalog\_category.assignedPro...  
catalog\_category.assignProduct  
catalog\_category.create  
catalog\_category.currentStore  
catalog\_category.delete  
catalog\_category.info  
catalog\_category.level  
catalog\_category.move  
catalog\_category.removeProduct  
[catalog\\_category.tree](#)  
catalog\_category.update  
catalog\_category.updateProduct

### Catalog Category Attributes ^

### Catalog Product ^

#### Catalog Product Attribute ^

#### Catalog Product Attribute

### Media ^

#### Catalog Product Attribute Set ^

#### Catalog Product Custom

#### Option ^

#### Catalog Product Custom Option

#### Value ^

#### Catalog Product Downloadable

#### Link ^

#### Catalog Product Link ^

#### Catalog Product Tag ^

#### Catalog Product Tier Price ^

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

Resource Name: catalog\_category

### Aliases:

- category

### Method:

- catalog\_category.tree (SOAP V1)
- catalogCategoryTree (SOAP V2)

Allows you to retrieve the hierarchical tree of categories.

### Aliases:

- category.tree

### Arguments:

| Type   | Name      | Description                          |
|--------|-----------|--------------------------------------|
| string | sessionId | Session ID                           |
| string | parentId  | ID of the parent category (optional) |
| string | storeView | Store view (optional)                |

### Returns:

| Type  | Name | Description                  |
|-------|------|------------------------------|
| array | tree | Array of catalogCategoryTree |

## Catalog Product Types ▾

The **catalogCategoryTree** content is as follows:

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

| Type   | Name        | Description                      |
|--------|-------------|----------------------------------|
| int    | category_id | Category ID                      |
| int    | parent_id   | Parent category ID               |
| string | name        | Category name                    |
| int    | position    | Category position                |
| int    | level       | Category level                   |
| array  | children    | Array of CatalogCategoryEntities |

The **catalogCategoryEntity** content is as follows:

| Type   | Name        | Description                            |
|--------|-------------|----------------------------------------|
| int    | category_id | Category ID                            |
| int    | parent_id   | Parent category ID                     |
| string | name        | Category name                          |
| int    | is_active   | defines whether the category is active |
| int    | position    | Category position                      |
| int    | level       | Category level                         |
| array  | children    | Array of CatalogCategoryEntities       |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category.tree');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryTree($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogCategoryTree((object)array('sessionId' => $sessionId->result,
'parentId' => '15'));
var_dump($result->result);

```

## Response Example SOAP V1

```

array
 'category_id' => string '1' (length=1)
 'parent_id' => string '0' (length=1)
 'name' => string 'Root Catalog' (length=12)
 'is_active' => null
 'position' => string '0' (length=1)
 'level' => string '0' (length=1)
 'children' =>
 array
 0 =>
 array
 'category_id' => string '2' (length=1)
 'parent_id' => string '1' (length=1)
 'name' => string 'Default Category' (length=16)
 'is_active' => string '1' (length=1)
 'position' => string '1' (length=1)
 'level' => string '1' (length=1)
 'children' =>
 array
 ...
 1 =>
 array
 'category_id' => string '3' (length=1)
 'parent_id' => string '1' (length=1)
 'name' => string 'root_category' (length=13)
 'is_active' => string '1' (length=1)
 'position' => string '2' (length=1)
 'level' => string '1' (length=1)
 'children' =>
 array
 ...

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Category Update

[Edit this page on GitHub](#) 

### Introduction

### Catalog

#### Catalog Category

- catalog\_category.assignedPro...
- catalog\_category.assignProduct
- catalog\_category.create
- catalog\_category.currentStore
- catalog\_category.delete
- catalog\_category.info
- catalog\_category.level
- catalog\_category.move
- catalog\_category.removeProduct
- catalog\_category.tree
- catalog\_category.update**
- catalog\_category.updateProduct

#### Catalog Category Attributes

#### Catalog Product

#### Catalog Product Attribute

#### Catalog Product Attribute

#### Media

#### Catalog Product Attribute Set

#### Catalog Product Custom

#### Option

#### Catalog Product Custom Option Value

#### Catalog Product Downloadable

#### Link

#### Catalog Product Link

#### Catalog Product Tag

#### Catalog Product Tier Price

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Resource Name: catalog\_category

#### Aliases:

- category

#### Method:

- catalog\_category.update (SOAP V1)
- catalogCategoryUpdate (SOAP V2)

Update the required category. Note that you should specify only those parameters which you want to be updated.

#### Aliases:

- category.update

#### Arguments:

| Type   | Name         | Description                             |
|--------|--------------|-----------------------------------------|
| string | sessionId    | Session ID                              |
| int    | categoryId   | ID of the category to be updated        |
| array  | categoryData | An array of catalogCategoryEntityCreate |
| string | storeView    | Store view ID or code (optional)        |

#### Returns:

| Type | Description |
|------|-------------|
|------|-------------|

**Catalog Product Types** ▾

|         |                                 |  |
|---------|---------------------------------|--|
| boolean | True if the category is updated |  |
|---------|---------------------------------|--|

**Catalog Inventory** ▾**Checkout** ▾**Create Your Own API****Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

The **catalogCategoryEntityCreate** content is as follows:

| Type          | Name                       | Description                                                                                                               |
|---------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------|
| string        | name                       | Name of the category to be updated                                                                                        |
| int           | is_active                  | Defines whether the category is visible in the frontend                                                                   |
| int           | position                   | Position of the category to be updated                                                                                    |
| arrayOfString | available_sort_by          | All available options by which products in the category can be sorted                                                     |
| string        | custom_design              | The custom design for the category                                                                                        |
| int           | custom_apply_to_products   | Apply the custom design to all products assigned to the category                                                          |
| string        | custom_design_from         | Date starting from which the custom design will be applied to the category                                                |
| string        | custom_design_to           | Date till which the custom design will be applied to the category                                                         |
| string        | custom_layout_update       | Custom layout update                                                                                                      |
| string        | default_sort_by            | The default option by which products in the category are sorted                                                           |
| string        | description                | Category description                                                                                                      |
| string        | display_mode               | Content that will be displayed on the category view page                                                                  |
| int           | is_anchor                  | Defines whether the category will be anchored                                                                             |
| int           | landing_page               | Landing page                                                                                                              |
| string        | meta_description           | Category meta description                                                                                                 |
| string        | meta_keywords              | Category meta keywords                                                                                                    |
| string        | meta_title                 | Category meta title                                                                                                       |
| string        | page_layout                | Type of page layout that the category should use                                                                          |
| string        | url_key                    | A relative URL path which can be entered in place of the standard target path                                             |
| int           | include_in_menu            | Defines whether the category is visible on the top menu bar in the frontend                                               |
| string        | filter_price_range         | Price range of each price level displayed in the layered navigation block                                                 |
| int           | custom_use_parent_settings | Defines whether the category will inherit custom design settings of the category to which it is assigned. 1 - Yes, 0 - No |

**Faults:**

No Faults

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
```

```

$session = $client->login('apiUser', 'apiKey');
$result = $client->call($session, 'catalog_category.update', array(23, array(
 'name' => 'Category name',
 'is_active' => 1,
 'position' => 1,
 //<!-- position parameter is deprecated, category anyway will be positioned in the
 end of list
 //and you can not set position directly, use catalog_category.move instead -->
 'available_sort_by' => 'position',
 'custom_design' => null,
 'custom_apply_to_products' => null,
 'custom_design_from' => null,
 'custom_design_to' => null,
 'custom_layout_update' => null,
 'default_sort_by' => 'position',
 'description' => 'Category description',
 'display_mode' => null,
 'is_anchor' => 0,
 'landing_page' => null,
 'meta_description' => 'Category meta description',
 'meta_keywords' => 'Category meta keywords',
 'meta_title' => 'Category meta title',
 'page_layout' => 'two_columns_left',
 'url_key' => 'url-key',
 'include_in_menu' => 1,
));
var_dump ($result);
</pre>

```

## Request Example SOAP V2

```

$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->catalogCategoryUpdate($session, 23, array(
 'name' => 'Category name 2',
 'is_active' => 1,
 'position' => 1,
 //<!-- position parameter is deprecated, category anyway will be positioned in the
 end of list
 //and you can not set position directly, use catalog_category.move instead -->
 'available_sort_by' => array('position'),
 'custom_design' => null,
 'custom_apply_to_products' => null,
 'custom_design_from' => null,
 'custom_design_to' => null,
 'custom_layout_update' => null,
 'default_sort_by' => 'position',
 'description' => 'Category description',
 'display_mode' => null,
 'is_anchor' => 0,
 'landing_page' => null,
 'meta_description' => 'Category meta description',
 'meta_keywords' => 'Category meta keywords',
 'meta_title' => 'Category meta title',
 'page_layout' => 'two_columns_left',
 'url_key' => 'url-key',
 'include_in_menu' => 1,
));
var_dump ($result);
</pre>

```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogCategoryUpdate((object)array('sessionId' => $sessionId-
>result, 'categoryId' => '23', 'categoryData' => ((object)array(
 'name' => 'Category Name Updated',
 'is_active' => '1',
 'position' => '1',
 'available_sort_by' => array('name'),
 'default_sort_by' => 'name',
 'description' => 'Category description',
 'is_anchor' => '1',
 'include_in_menu' => '1'
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Category Product Update

[Edit this page on GitHub](#) 

## Introduction

## Catalog

### Catalog Category

- catalog\_category.assignedPro...
- catalog\_category.assignProduct
- catalog\_category.create
- catalog\_category.currentStore
- catalog\_category.delete
- catalog\_category.info
- catalog\_category.level
- catalog\_category.move
- catalog\_category.removeProduct
- catalog\_category.tree
- catalog\_category.update
- [catalog\\_category.updateProduct](#)

### Catalog Category Attributes

### Catalog Product

#### Catalog Product Attribute

#### Catalog Product Attribute

## Media

### Catalog Product Attribute Set

### Catalog Product Custom

#### Option

### Catalog Product Custom Option

#### Value

### Catalog Product Downloadable

#### Link

### Catalog Product Link

### Catalog Product Tag

### Catalog Product Tier Price

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

**Resource Name:** catalog\_category

### Aliases:

- category

### Method:

- catalog\_category.updateProduct (SOAP V1)
- catalogCategoryUpdateProduct (SOAP V2)

Allows you to update the product assigned to a category. The product position is updated.

### Aliases:

- category.updateProduct

### Arguments:

| Type   | Name           | Description                                                                |
|--------|----------------|----------------------------------------------------------------------------|
| string | sessionId      | Session ID                                                                 |
| int    | categoryId     | ID of the category to which the product is assigned                        |
| string | productId      | ID or SKU of the product to be updated                                     |
| string | position       | Position of the product in the category (optional)                         |
| string | identifierType | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

## Catalog Product Types ▾

| Type    | Description                                    |
|---------|------------------------------------------------|
| boolean | True if the product is updated in the category |

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category.updateProduct', array('categoryId' => '4', 'productId' => '1', 'position' => '3'));
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryUpdateProduct($sessionId, '4', '1', '3');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogCategoryUpdateProduct((object)array('sessionId' => $sessionId->result, 'categoryId' => '4', 'productId' => '1', 'position' => '3'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Category Attributes

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

### Catalog Category ▼

#### Catalog Category Attributes ^

- catalog\_category\_attribute.cur...
- catalog\_category\_attribute.list
- catalog\_category\_attribute.opti...

### Catalog Product ▼

#### Catalog Product Attribute ▼

##### Catalog Product Attribute

##### Media ▼

#### Catalog Product Attribute Set ▼

##### Catalog Product Custom

##### Option ▼

##### Catalog Product Custom Option

##### Value ▼

##### Catalog Product Downloadable

##### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

## Catalog Inventory ▼

## Checkout ▼

## Create Your Own API

## Customer ▼

## Directory ▼

## Sales ▼

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Category Attributes

Allows you to retrieve the list of category attributes and options.

**Resource Name:** catalog\_category\_attribute

#### Aliases:

- category\_attribute

#### Methods:

- [catalog\\_category\\_attribute.currentStore](#) - Set/Get the current store view
- [catalog\\_category\\_attribute.list](#) - Retrieve the category attributes
- [catalog\\_category\\_attribute.options](#) - Retrieve the attribute options

### Faults

| Fault Code | Fault Message                   |
|------------|---------------------------------|
| 100        | Requested store view not found. |
| 101        | Requested attribute not found.  |

### Examples

#### Example 1. Retrieving attributes and options

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
```

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

---

```
$attributes = $proxy->call($sessionId, 'category_attribute.list');
foreach ($attributes as &$attribute) {
 if (isset($attribute['type']))
 && ($attribute['type'] == 'select' || $attribute['type'] == 'multiselect')) {
 $attribute['options'] = $proxy->call($sessionId, 'category_attribute.options',
$attribute['code']);
 }
}
var_dump($attributes);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Current Store

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ^

- [catalog\\_category\\_attribute.cur...](#)
- [catalog\\_category\\_attribute.list](#)
- [catalog\\_category\\_attribute.opti...](#)

#### Catalog Product ▼

#### Catalog Product Attribute ▼

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

#### Catalog Inventory ▼

### Checkout ▼

### Create Your Own API

### Customer ▼

### Directory ▼

### Sales ▼

## Module: Mage\_Catalog

Resource: catalog\_category\_attribute

#### Aliases:

- category\_attribute

#### Method:

- catalog\_category\_attribute.currentStore (SOAP V1)
- catalogCategoryAttributeCurrentStore (SOAP V2)

Allows you to set/get the current store view.

#### Aliases:

- category\_attribute.currentStore

#### Arguments:

| Type   | Name      | Description           |
|--------|-----------|-----------------------|
| string | sessionId | Session ID            |
| string | storeView | Store view ID or code |

#### Returns:

| Type | Name      | Description   |
|------|-----------|---------------|
| int  | storeView | Store view ID |

## Examples

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category_attribute.currentStore',
'english');
var_dump ($result);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if
necessary

$result = $proxy->catalogCategoryAttributeCurrentStore($sessionId, 'english');
var_dump($result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▲

- catalog\_category\_attribute.cur...
- [catalog\\_category\\_attribute.list](#)
- catalog\_category\_attribute.opti...

#### Catalog Product ▼

#### Catalog Product Attribute ▼

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

### Catalog Inventory ▼

### Checkout ▼

### Create Your Own API

### Customer ▼

### Directory ▼

### Sales ▼

## Module: Mage\_Catalog

Resource: catalog\_category\_attribute

#### Aliases:

- category\_attribute

#### Method:

- catalog\_category\_attribute.list (SOAP V1)
- catalogCategoryAttributeList (SOAP V2)

Allows you to retrieve the list of category attributes.

#### Aliases:

- category\_attribute.list

#### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

#### Returns:

| Type  | Name   | Description                     |
|-------|--------|---------------------------------|
| array | result | Array of catalogAttributeEntity |

The **catalogAttributeEntity** content is as follows:

| Type | Name         | Description  |
|------|--------------|--------------|
| int  | attribute_id | Attribute ID |

**Enterprise Customer Balance** ✓

**Enterprise Gift Card** ✓

**Enterprise Gift Message** ✓

**Miscellaneous** ✓

**WS-I Compliance**

|  |        |          |                                            |
|--|--------|----------|--------------------------------------------|
|  | string | code     | Attribute code                             |
|  | string | type     | Attribute type                             |
|  | string | required | Defines whether the attribute is required  |
|  | string | scope    | Attribute scope: global, website, or store |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category_attribute.list');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryAttributeList($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$session = $client->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $client->catalogCategoryAttributeList((object)array('sessionId' => $session->result));

var_dump ($result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'attribute_id' => null
 'code' => string 'parent_id' (length=9)
 'type' => null
 'required' => null
 'scope' => string 'global' (length=6)
1 =>
array
 'attribute_id' => null
 'code' => string 'increment_id' (length=12)
```

```
'type' => null
'required' => null
'scope' => string 'global' (length=6)
2 =>
array
'attribute_id' => null
'code' => string 'updated_at' (length=10)
'type' => null
'required' => null
'scope' => string 'global' (length=6)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Attribute Options

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▲

[catalog\\_category\\_attribute.cur...](#)

[catalog\\_category\\_attribute.list](#)

[catalog\\_category\\_attribute.opti...](#)

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

[Sales](#) ▼

## Module: Mage\_Catalog

Resource: catalog\_category\_attribute

### Aliases:

- category\_attribute

### Method:

- catalog\_category\_attribute.options (SOAP V1)
- catalogCategoryAttributeOptions (SOAP V2)

Allows you to retrieve the attribute options.

### Aliases:

- category\_attribute.options

### Arguments:

| Type   | Name        | Description           |
|--------|-------------|-----------------------|
| string | sessionId   | Session ID            |
| string | attributeId | Attribute ID or code  |
| string | storeView   | Store view ID or code |

### Returns:

| Type  | Name   | Description                           |
|-------|--------|---------------------------------------|
| array | result | Array of catalogAttributeOptionEntity |

The **catalogAttributeOptionEntity** content is as follows:

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

| Type   | Name  | Description  |
|--------|-------|--------------|
| string | label | Option label |
| string | value | Option value |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_category_attribute.options', '65');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogCategoryAttributeOptions($sessionId, '65');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$session = $client->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $client->catalogCategoryAttributeOptions((object)array('sessionId' => $session->result, 'attributeId' => '65'));

var_dump ($result);
```

### Response Example SOAP V1

```
array
0 =>
 array
 'label' => string 'Yes' (length=3)
 'value' => int 1
1 =>
 array
 'label' => string 'No' (length=2)
 'value' => int 0
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▼

#### Catalog Product ▲

`catalog_product.create`  
`catalog_product.currentStore`  
`catalog_product.delete`  
`catalog_product.getSpecialPrice`  
`catalog_product.info`  
`catalog_product.list`  
`catalog_product.listOfAdditionalAttributes`  
`catalog_product.setSpecialPrice`  
`catalog_product.update`

#### Catalog Product Attribute ▼

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

#### Catalog Inventory ▼

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Product

Allows you to manage products.

**Resource Name:** catalog\_product

#### Aliases:

- product

#### Methods:

- `catalog_product.currentStore` - Set/Get the current store view
- `catalog_product.list` - Retrieve the list of products using filters
- `catalog_product.info` - Retrieve information about the required product
- `catalog_product.create` - Create a new product
- `catalog_product.update` - Update a required product
- `catalog_product.setSpecialPrice` - Set special price for a product
- `catalog_product.getSpecialPrice` - Get special price for a product
- `catalog_product.delete` - Delete a required product
- `catalog_product.listOfAdditionalAttributes` - Get the list of additional attributes

### Faults

| Fault Code | Fault Message                   |
|------------|---------------------------------|
| 100        | Requested store view not found. |
| 101        | Product not exists.             |

[Checkout](#) ▾  
[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

102 Invalid data given. Details in error message.

103 Product not deleted. Details in error message.

## Examples

### Example 1. Retrieving the product list

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$filters = array(
 'sku' => array('like'=>'zol%')
);

$products = $proxy->call($sessionId, 'product.list', array($filters));

var_dump($products);
```

### Example 2. Creating, viewing, updating, and deleting a product

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$attributeSets = $proxy->call($sessionId, 'product_attribute_set.list');
$set = current($attributeSets);

$newProductData = array(
 'name' => 'name of product',
 // websites - Array of website ids to which you want to assign a new product
 'websites' => array(1), // array(1,2,3,...)
 'short_description' => 'short description',
 'description' => 'description',
 'status' => 1,
 'weight' => 0,
 'tax_class_id' => 1,
 'categories' => array(3), //3 is the category id
 'price' => 12.05
);

// Create new product
$proxy->call($sessionId, 'product.create', array('simple', $set['set_id'],
'sku_of_product', $newProductData));
$proxy->call($sessionId, 'product_stock.update', array('sku_of_product',
array('qty'=>50, 'is_in_stock'=>1)));

// Get info of created product
var_dump($proxy->call($sessionId, 'product.info', 'sku_of_product'));

// Update product name on german store view
$proxy->call($sessionId, 'product.update', array('sku_of_product', array('name'=>'new
name of product'), 'german'));

// Get info for default values
var_dump($proxy->call($sessionId, 'product.info', 'sku_of_product'));
// Get info for german store view

var_dump($proxy->call($sessionId, 'product.info', array('sku_of_product', 'german')));
```

```
// Delete product
$proxy->call($sessionId, 'product.delete', 'sku_of_product');

try {
 // Ensure that product deleted
 var_dump($proxy->call($sessionId, 'product.info', 'sku_of_product'));
} catch (SoapFault $e) {
 echo "Product already deleted";
}
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Create

[Edit this page on GitHub](#) 

## Introduction

## Catalog

### Catalog Category

### Catalog Category Attributes

### Catalog Product

#### `catalog_product.create`

`catalog_product.currentStore`

`catalog_product.delete`

`catalog_product.getSpecialPrice`

`catalog_product.info`

`catalog_product.list`

`catalog_product.listOfAddition...`

`catalog_product.setSpecialPrice`

`catalog_product.update`

### Catalog Product Attribute

### Catalog Product Attribute

## Media

### Catalog Product Attribute Set

### Catalog Product Custom

### Option

### Catalog Product Custom Option

### Value

### Catalog Product Downloadable

### Link

### Catalog Product Link

### Catalog Product Tag

### Catalog Product Tier Price

### Catalog Product Types

### Catalog Inventory

## Module: Mage\_Catalog

Resource: catalog\_product

### Aliases:

- product

### Method:

- `catalog_product.create` (SOAP V1)
- `catalogProductCreate` (SOAP V2)

Allows you to create a new product and return ID of the created product.

### Aliases:

- `product.create`

### Note:

Although the API accepts up to four digits of precision for all price arguments, Magento strongly recommends you pass in two digits to reduce inaccuracy in the tax calculation process. (That is, use a price like 12.35 and not 12.3487).

### Arguments:

| Type   | Name      | Description                     |
|--------|-----------|---------------------------------|
| string | sessionId | Session ID                      |
| string | type      | Product type                    |
| string | set       | ID of the product attribute set |
| string | sku       | Product SKU                     |

**Checkout** ▾**Create Your Own API****Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾

|        |             |                                     |
|--------|-------------|-------------------------------------|
| array  | productData | Array of catalogProductCreateEntity |
| string | storeView   | Store view ID or code               |

**Returns:**

| Type | Name   | Description               |
|------|--------|---------------------------|
| int  | result | ID of the created product |

The **catalogProductCreateEntity** content is as follows:

| Type          | Name                   | Description                                                               |
|---------------|------------------------|---------------------------------------------------------------------------|
| ArrayOfString | categories             | Array of categories                                                       |
| ArrayOfString | websites               | Array of websites                                                         |
| string        | name                   | Product name                                                              |
| string        | description            | Product description                                                       |
| string        | short_description      | Product short description                                                 |
| string        | weight                 | Product weight                                                            |
| string        | status                 | Product status                                                            |
| string        | url_key                | URL key                                                                   |
| string        | url_path               | URL path                                                                  |
| string        | visibility             | Product visibility on the frontend                                        |
| ArrayOfString | category_ids           | Array of category IDs                                                     |
| ArrayOfString | website_ids            | Array of website IDs                                                      |
| string        | has_options            | Defines whether the product has options                                   |
| string        | gift_message_available | Defines whether the gift message is available for the product             |
| string        | price                  | Product price                                                             |
| string        | special_price          | Product special price                                                     |
| string        | special_from_date      | Date starting from which the special price will be applied to the product |
| string        | special_to_date        | Date till which the special price will be applied to the product          |
| string        | tax_class_id           | Tax class ID                                                              |
| array         | tier_price             | Array of catalogProductTierPriceEntity                                    |
| string        | meta_title             | Meta title                                                                |
| string        | meta_keyword           | Meta keyword                                                              |
| string        | meta_description       | Meta description                                                          |
| string        | custom_design          | Custom design                                                             |
| string        | custom_layout_update   | Custom layout update                                                      |

|        |                       |                                                   |
|--------|-----------------------|---------------------------------------------------|
| string | options_container     | Options container                                 |
| array  | additional_attributes | Array of catalogProductAdditionalAttributesEntity |
| array  | stock_data            | Array of catalogInventoryStockItemUpdateEntity    |

**Notes:** The "websites" and "website\_ids" or "categories" and "category\_ids" parameters are interchangeable. In other words, you can specify an array of website IDs (int) and then you don't need to specify the array of website codes (string) and vice versa.

The **catalogProductTierPriceEntity** content is as follows:

| Type   | Name              | Description       |
|--------|-------------------|-------------------|
| string | customer_group_id | Customer group ID |
| string | website           | Website           |
| int    | qty               | Quantity          |
| double | price             | Tier price        |

The **catalogInventoryStockItemUpdateEntity** content is as follows:

| Type   | Name                        | Description                                                                    |
|--------|-----------------------------|--------------------------------------------------------------------------------|
| string | qty                         | Quantity of items                                                              |
| int    | is_in_stock                 | Defines whether the item is in stock                                           |
| int    | manage_stock                | Manage stock                                                                   |
| int    | use_config_manage_stock     | Use config manage stock                                                        |
| int    | min_qty                     | Minimum quantity for items to be in stock                                      |
| int    | use_config_min_qty          | Use config settings flag (value defined in the Inventory System Configuration) |
| int    | min_sale_qty                | Minimum quantity allowed in the shopping cart                                  |
| int    | use_config_min_sale_qty     | Use config settings flag                                                       |
| int    | max_sale_qty                | Maximum quantity allowed in the shopping cart                                  |
| int    | use_config_max_sale_qty     | Use config settings flag                                                       |
| int    | is_qty_decimal              | Defines whether the quantity is decimal                                        |
| int    | backorders                  | Backorders status                                                              |
| int    | use_config_backorders       | Use config settings flag (for backorders)                                      |
| int    | notify_stock_qty            | Stock quantity below which a notification will appear                          |
| int    | use_config_notify_stock_qty | Use config settings flag (for stock quantity)                                  |

The **catalogProductAdditionalAttributesEntity** content is as follows:

| Type                  | Name        |
|-----------------------|-------------|
| associativeMultiArray | multi_data  |
| associativeArray      | single_data |

Single Data: array of attributes with only single value  
Multi Data: array of attributes which could contain several values

#### Faults:

| Fault Code | Fault Message                                                   |
|------------|-----------------------------------------------------------------|
| 100        | Requested store view not found.                                 |
| 102        | Invalid data given. Details in error message.                   |
| 104        | Product type is not in allowed types.                           |
| 105        | Product attribute set is not existed                            |
| 106        | Product attribute set is not belong catalog product entity type |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

// get attribute set
$attributeSets = $client->call($session, 'product_attribute_set.list');
$attributeSet = current($attributeSets);

$result = $client->call($session, 'catalog_product.create', array('simple',
$attributeSet['set_id'], 'product_sku', array(
 'categories' => array(2),
 'websites' => array(1),
 'name' => 'Product name',
 'description' => 'Product description',
 'short_description' => 'Product short description',
 'weight' => '10',
 'status' => '1',
 'url_key' => 'product-url-key',
 'url_path' => 'product-url-path',
 'visibility' => '4',
 'price' => '100',
 'tax_class_id' => 1,
 'meta_title' => 'Product meta title',
 'meta_keyword' => 'Product meta keyword',
 'meta_description' => 'Product meta description'
));
var_dump ($result);
```

### Request Example SOAP V2

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
```

```

// get attribute set
$attributeSets = $client->catalogProductAttributeSetList($session);
$attributeSet = current($attributeSets);

$result = $client->catalogProductCreate($session, 'simple', $attributeSet->set_id,
'product_sku', array(
 'categories' => array(2),
 'websites' => array(1),
 'name' => 'Product name',
 'description' => 'Product description',
 'short_description' => 'Product short description',
 'weight' => '10',
 'status' => '1',
 'url_key' => 'product-url-key',
 'url_path' => 'product-url-path',
 'visibility' => '4',
 'price' => '100',
 'tax_class_id' => 1,
 'meta_title' => 'Product meta title',
 'meta_keyword' => 'Product meta keyword',
 'meta_description' => 'Product meta description'
));
var_dump ($result);

```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductCreate((object)array('sessionId' => $sessionId-
>result, 'type' => 'simple', 'set' => '4', 'sku' => 'simple_sku',
'productData' => ((object)array(
 'name' => 'Product name',
 'description' => 'Product description',
 'short_description' => 'Product short description',
 'weight' => '10',
 'status' => '1',
 'visibility' => '4',
 'price' => '100',
 'tax_class_id' => 1,
))));

var_dump($result->result);

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Current Product Store

[Edit this page on GitHub](#) 

### Introduction

### Catalog

#### Catalog Category

#### Catalog Category Attributes

#### Catalog Product

- catalog\_product.create
- catalog\_product.currentStore
- catalog\_product.delete
- catalog\_product.getSpecialPrice
- catalog\_product.info
- catalog\_product.list
- catalog\_product.listOfAddition...
- catalog\_product.setSpecialPrice
- catalog\_product.update

#### Catalog Product Attribute

#### Catalog Product Attribute

#### Media

#### Catalog Product Attribute Set

#### Catalog Product Custom

#### Option

#### Catalog Product Custom Option

#### Value

#### Catalog Product Downloadable

#### Link

#### Catalog Product Link

#### Catalog Product Tag

#### Catalog Product Tier Price

#### Catalog Product Types

#### Catalog Inventory

## Module: Mage\_Catalog

Resource: catalog\_product

#### Aliases:

- product

#### Method:

- catalog\_product.currentStore (SOAP V1)
- catalogProductCurrentStore (SOAP V2)

Allows you to set/get the current store view.

#### Aliases:

- product.currentStore

#### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | storeView | Store view ID or code (optional) |

#### Returns:

| Type       | Name      | Description   |
|------------|-----------|---------------|
| int/string | storeView | Store view ID |

## Examples

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product.currentStore', 'english');
var_dump ($result);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductCurrentStore($sessionId, 'english');
var_dump($result);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Delete

[Edit this page on GitHub](#) 

## Introduction

## Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

- [catalog\\_product.create](#)
- [catalog\\_product.currentStore](#)
- [catalog\\_product.delete](#)
- [catalog\\_product.getSpecialPrice](#)
- [catalog\\_product.info](#)
- [catalog\\_product.list](#)
- [catalog\\_product.listOfAddition...](#)
- [catalog\\_product.setSpecialPrice](#)
- [catalog\\_product.update](#)

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 



## Module: Mage\_Catalog

Resource: catalog\_product

### Aliases:

- product

### Method:

- catalog\_product.delete (SOAP V1)
- catalogProductDelete (SOAP V2)

Allows you to delete the required product.

### Aliases:

- product.delete

### Arguments:

| Type   | Name              | Description                                                                |
|--------|-------------------|----------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                 |
| string | product\productId | Product ID or SKU                                                          |
| string | identifierType    | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

| Type        | Name   | Description                        |
|-------------|--------|------------------------------------|
| boolean/int | result | True (1) if the product is deleted |

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product.delete', '6');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductDelete($sessionId, '6');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductDelete((object)array('sessionId' => $sessionId->result, 'productId' => '21'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Get Special Price

[Edit this page on GitHub](#) 

### Introduction

### Catalog

#### Catalog Category

#### Catalog Category Attributes

#### Catalog Product

- catalog\_product.create
- catalog\_product.currentStore
- catalog\_product.delete
- catalog\_product.getSpecialPrice**
- catalog\_product.info
- catalog\_product.list
- catalog\_product.listOfAddition...
- catalog\_product.setSpecialPrice
- catalog\_product.update

#### Catalog Product Attribute

#### Catalog Product Attribute

#### Media

#### Catalog Product Attribute Set

#### Catalog Product Custom

#### Option

#### Catalog Product Custom Option

#### Value

#### Catalog Product Downloadable

#### Link

#### Catalog Product Link

#### Catalog Product Tag

#### Catalog Product Tier Price

#### Catalog Product Types

#### Catalog Inventory

## Module: Mage\_Catalog

Resource: catalog\_product

#### Aliases:

- product

#### Method:

- catalog\_product.getSpecialPrice (SOAP V1)
- catalogProductGetSpecialPrice (SOAP V2)

Allows you to get the product special price data.

#### Aliases:

- product.getSpecialPrice

#### Arguments:

| Type   | Name              | Description                                                                |
|--------|-------------------|----------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                 |
| string | product\productId | Product ID or SKU                                                          |
| string | storeView         | Store view ID or code                                                      |
| string | identifierType    | Defines whether the product ID or SKU is passed in the 'product' parameter |

#### Returns:

| Type  | Name   | Description                         |
|-------|--------|-------------------------------------|
| array | result | Array of catalogProductReturnEntity |

[Checkout](#) ▾[Create Your Own API](#)[Customer](#) ▾[Directory](#) ▾[Sales](#) ▾[Enterprise Customer Balance](#) ▾[Enterprise Gift Card](#) ▾[Enterprise Gift Message](#) ▾[Miscellaneous](#) ▾[WS-I Compliance](#)

The **catalogProductReturnEntity** content is as follows:

| Type   | Name              | Description                                                          |
|--------|-------------------|----------------------------------------------------------------------|
| string | special_price     | Product special price                                                |
| string | special_from_date | Date starting from which the special price is applied to the product |
| string | special_to_date   | Date till which the special price is applied to the product          |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product.getSpecialPrice', '1');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductGetSpecialPrice($sessionId, '1');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductGetSpecialPrice((object)array('sessionId' => $sessionId->result, 'productId' => '1'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 'special_price' => string '139.9900' (length=8)
 'special_from_date' => string '2012-03-28 00:00:00' (length=19)
 'special_to_date' => string '2012-03-31 00:00:00' (length=19)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog

#### Catalog Category

#### Catalog Category Attributes

#### Catalog Product

- catalog\_product.create
- catalog\_product.currentStore
- catalog\_product.delete
- catalog\_product.getSpecialPrice
- catalog\_product.info**
- catalog\_product.list
- catalog\_product.listOfAddition...
- catalog\_product.setSpecialPrice
- catalog\_product.update

#### Catalog Product Attribute

#### Catalog Product Attribute

#### Media

#### Catalog Product Attribute Set

#### Catalog Product Custom

#### Option

#### Catalog Product Custom Option

#### Value

#### Catalog Product Downloadable

#### Link

#### Catalog Product Link

#### Catalog Product Tag

#### Catalog Product Tier Price

#### Catalog Product Types

## Module: Mage\_Catalog

Resource: catalog\_product

#### Aliases:

- product

#### Method:

- catalog\_product.info (SOAP V1)
- catalogProductInfo (SOAP V2)

Allows you to retrieve information about the required product.

#### Aliases:

- product.info

#### Arguments:

| Type   | Name           | Description                                                                                    |
|--------|----------------|------------------------------------------------------------------------------------------------|
| string | sessionId      | Session ID                                                                                     |
| string | productId      | Product ID or SKU                                                                              |
| string | storeView      | Store view ID or code (optional)                                                               |
| array  | attributes     | Array of catalogProductRequestAttributes (optional)                                            |
| string | identifierType | Defines whether the product ID or SKU value is passed in the "productId" parameter. (optional) |

#### Returns:

| Type | Name | Description |
|------|------|-------------|
|      |      |             |

[Checkout](#) ▾[Create Your Own API](#)[Customer](#) ▾[Directory](#) ▾[Sales](#) ▾[Enterprise Customer Balance](#) ▾[Enterprise Gift Card](#) ▾[Enterprise Gift Message](#) ▾[Miscellaneous](#) ▾

|       |      |                                     |
|-------|------|-------------------------------------|
| array | info | Array of catalogProductReturnEntity |
|-------|------|-------------------------------------|

The **catalogProductRequestAttributes** content is as follows:

| Type          | Name                  | Description                    |
|---------------|-----------------------|--------------------------------|
| ArrayOfString | attributes            | Array of attributes            |
| ArrayOfString | additional_attributes | Array of additional attributes |

The **catalogProductReturnEntity** content is as follows:

| Type          | Name                   | Description                                                          |
|---------------|------------------------|----------------------------------------------------------------------|
| string        | product_id             | Product ID                                                           |
| string        | sku                    | Product SKU                                                          |
| string        | set                    | Product set                                                          |
| string        | type                   | Product type                                                         |
| ArrayOfString | categories             | Array of categories                                                  |
| ArrayOfString | websites               | Array of websites                                                    |
| string        | created_at             | Date when the product was created                                    |
| string        | updated_at             | Date when the product was last updated                               |
| string        | type_id                | Type ID                                                              |
| string        | name                   | Product name                                                         |
| string        | description            | Product description                                                  |
| string        | short_description      | Short description for a product                                      |
| string        | weight                 | Product weight                                                       |
| string        | status                 | Status of a product                                                  |
| string        | url_key                | Relative URL path that can be entered in place of a target path      |
| string        | url_path               | URL path                                                             |
| string        | visibility             | Product visibility on the frontend                                   |
| ArrayOfString | category_ids           | Array of category IDs                                                |
| ArrayOfString | website_ids            | Array of website IDs                                                 |
| string        | has_options            | Defines whether the product has options                              |
| string        | gift_message_available | Defines whether the gift message is available for the product        |
| string        | price                  | Product price                                                        |
| string        | special_price          | Product special price                                                |
| string        | special_from_date      | Date starting from which the special price is applied to the product |
| string        | special_to_date        | Date till which the special price is applied to the product          |

|                  |                       |                                                           |
|------------------|-----------------------|-----------------------------------------------------------|
| string           | tax_class_id          | Tax class ID                                              |
| array            | tier_price            | Array of catalogProductTierPriceEntity                    |
| string           | meta_title            | Mate title                                                |
| string           | meta_keyword          | Meta keyword                                              |
| string           | meta_description      | Meta description                                          |
| string           | custom_design         | Custom design                                             |
| string           | custom_layout_update  | Custom layout update                                      |
| string           | options_container     | Options container                                         |
| associativeArray | additional_attributes | Array of additional attributes                            |
| string           | enable_googlecheckout | Defines whether Google Checkout is applied to the product |

The **catalogProductTierPriceEntity** content is as follows:

| Type   | Name              | Description                                 |
|--------|-------------------|---------------------------------------------|
| string | customer_group_id | ID of the customer group                    |
| string | website           | Website                                     |
| int    | qty               | Quantity to which the price will be applied |
| double | price             | Price that each item will cost              |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product.info', '4');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductInfo($sessionId, '4');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
```

```

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductInfo((object)array('sessionId' => $sessionId->result,
'productId' => '4'));
var_dump($result->result);

```

## Response Example SOAP V1

```

array
 'product_id' => string '4' (length=1)
 'sku' => string 'canon_powershot' (length=15)
 'set' => string '4' (length=1)
 'type' => string 'simple' (length=6)
 'categories' =>
 array
 0 => string '3' (length=1)
 1 => string '5' (length=1)
 'websites' =>
 array
 0 => string '2' (length=1)
 'type_id' => string 'simple' (length=6)
 'name' => string 'Canon PowerShot A630 8MP Digital Camera with 4x Optical Zoom'
 (length=60)
 'description' => string 'Replacing the highly popular PowerShot A620, the PowerShot
A630 features a rotating 2.5-inch vari-angle LCD, 4x optical zoom lens, and a vast
array of creative shooting modes.

The PowerShot A630 packs a vast array of advanced features into a remarkably compact
space' (length=267)
 'short_description' => string 'Replacing the highly popular PowerShot A620, the
PowerShot A630 features a rotating 2.5-inch vari-angle LCD, 4x optical zoom lens, and a
vast array of creative shooting modes.' (length=175)
 'weight' => string '1.0000' (length=6)
 'old_id' => null
 'news_from_date' => null
 'news_to_date' => null
 'status' => string '1' (length=1)
 'url_key' => string 'canon-powershot-a630-8mp-digital-camera-with-4x-optical-zoom'
 (length=60)
 'url_path' => string 'canon-powershot-a630-8mp-digital-camera-with-4x-optical-
zoom.html' (length=65)
 'visibility' => string '4' (length=1)
 'category_ids' =>
 array
 0 => string '3' (length=1)
 1 => string '5' (length=1)
 'required_options' => string '0' (length=1)
 'has_options' => string '0' (length=1)
 'image_label' => null
 'small_image_label' => null
 'thumbnail_label' => null
 'created_at' => string '2012-03-29 12:47:56' (length=19)
 'updated_at' => string '2012-03-29 12:47:56' (length=19)
 'country_of_manufacture' => null
 'price' => string '329.9900' (length=8)
 'group_price' =>
 array
 empty
 'special_price' => null
 'special_from_date' => null
 'special_to_date' => null

```

```
'tier_price' =>
 array
 empty
'minimal_price' => null
'msrp_enabled' => string '2' (length=1)
'msrp_display_actual_price_type' => string '4' (length=1)
'msrp' => null
'enable_googlecheckout' => string '1' (length=1)
'tax_class_id' => string '2' (length=1)
'meta_title' => null
'meta_keyword' => null
'meta_description' => null
'is_recurring' => string '0' (length=1)
'recurring_profile' => null
'custom_design' => null
'custom_design_from' => null
'custom_design_to' => null
'custom_layout_update' => null
'page_layout' => null
'options_container' => string 'container2' (length=10)
'gift_message_available' => null
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product List

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

### Catalog Category ▼

### Catalog Category Attributes ▼

### Catalog Product ▲

- catalog\_product.create
- catalog\_product.currentStore
- catalog\_product.delete
- catalog\_product.getSpecialPrice
- catalog\_product.info
- [catalog\\_product.list](#)
- catalog\_product.listOfAddition...
- catalog\_product.setSpecialPrice
- catalog\_product.update

### Catalog Product Attribute ▼

### Catalog Product Attribute

## Media ▼

### Catalog Product Attribute Set ▼

### Catalog Product Custom

### Option ▼

### Catalog Product Custom Option

### Value ▼

### Catalog Product Downloadable

### Link ▼

### Catalog Product Link ▼

### Catalog Product Tag ▼

### Catalog Product Tier Price ▼

### Catalog Product Types ▼

## Catalog Inventory ▼

## Module: Mage\_Catalog

Resource: catalog\_product

### Aliases:

- product

### Method:

- catalog\_product.list (SOAP V1)
- catalogProductList (SOAP V2)

Allows you to retrieve the list of products.

### Aliases:

- product.list

### Arguments:

| Type   | Name      | Description                               |
|--------|-----------|-------------------------------------------|
| string | sessionId | Session ID                                |
| array  | filters   | Array of filters by attributes (optional) |
| string | storeView | Store view ID or code (optional)          |

### Returns:

| Type  | Name      | Description                   |
|-------|-----------|-------------------------------|
| array | storeView | Array of catalogProductEntity |

The **catalogProductEntity** content is as follows:

**Checkout** ▾[Create Your Own API](#)**Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

| Type          | Name         | Description           |
|---------------|--------------|-----------------------|
| string        | product_id   | Product ID            |
| string        | sku          | Product SKU           |
| string        | name         | Product name          |
| string        | set          | Product attribute set |
| string        | type         | Type of the product   |
| ArrayOfString | category_ids | Array of category IDs |
| ArrayOfString | website_ids  | Array of website IDs  |

## Examples

**Request Example SOAP V1**

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product.list');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

**Request Example SOAP V2 (List of All Products)**

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductList($sessionId);
var_dump($result);
```

**Request Example SOAP V2 (Complex Filter)**

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$complexFilter = array(
 'complex_filter' => array(
 array(
 'key' => 'type',
 'value' => array('key' => 'in', 'value' => 'simple,configurable')
)
)
);
$result = $client->catalogProductList($session, $complexFilter);

var_dump ($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductList((object)array('sessionId' => $sessionId->result,
'filters' => null));
var_dump($result->result);
```

## Response Example SOAP V1

```
array
0 =>
array
'product_id' => string '1' (length=1)
'sku' => string 'n2610' (length=5)
'name' => string 'Nokia 2610 Phone' (length=16)
'set' => string '4' (length=1)
'type' => string 'simple' (length=6)
'category_ids' =>
array
0 => string '4' (length=1)
1 =>
array
'product_id' => string '2' (length=1)
'sku' => string 'b8100' (length=5)
'name' => string 'BlackBerry 8100 Pearl' (length=21)
'set' => string '4' (length=1)
'type' => string 'simple' (length=6)
'category_ids' =>
array
0 => string '4' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## List of Additional Product Attributes

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▼

#### Catalog Product ▲

[catalog\\_product.create](#)  
[catalog\\_product.currentStore](#)  
[catalog\\_product.delete](#)  
[catalog\\_product.getSpecialPrice](#)  
[catalog\\_product.info](#)  
[catalog\\_product.list](#)  
[catalog\\_product.listOfAdditionalAttributes](#)  
[catalog\\_product.setSpecialPrice](#)  
[catalog\\_product.update](#)

#### Catalog Product Attribute ▼

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

#### Catalog Inventory ▼

### Module: Product API

#### Resource: product

##### Method:

- [product.listOfAdditionalAttributes \(SOAP V1\)](#)
- [catalogProductListOfAdditionalAttributes \(SOAP V2\)](#)

Get the list of additional attributes. Additional attributes are attributes that are not in the default set of attributes.

##### Arguments:

| Type   | Name           | Description                 |
|--------|----------------|-----------------------------|
| string | sessionId      | Session ID                  |
| string | productType    | Product type (e.g., simple) |
| string | attributeSetId | Attribute set ID            |

##### Returns:

| Type   | Name         | Description                                     |
|--------|--------------|-------------------------------------------------|
| int    | attribute_id | Attribute ID                                    |
| string | code         | Attribute code                                  |
| string | type         | Attribute type (e.g., text, select, date, etc.) |
| string | required     | Defines whether the attribute is required       |
| string | scope        | Attribute scope (global, website, or store)     |

##### Faults:

## Checkout ▾

### Create Your Own API

#### Customer ▾

#### Directory ▾

#### Sales ▾

#### Enterprise Customer Balance ▾

#### Enterprise Gift Card ▾

#### Enterprise Gift Message ▾

#### Miscellaneous ▾

#### WS-I Compliance

| Fault Code | Fault Message                                                   |
|------------|-----------------------------------------------------------------|
| 104        | Product type is not in allowed types.                           |
| 105        | Product attribute set is not existed                            |
| 106        | Product attribute set is not belong catalog product entity type |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$listAttributes = $proxy->call(
 $sessionId,
 'product.listOfAdditionalAttributes',
 array(
 'simple',
 13
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductListOfAdditionalAttributes($sessionId, 'simple',
'13');
var_dump($result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'attribute_id' => string '89' (length=2)
 'code' => string 'old_id' (length=6)
 'type' => string 'text' (length=4)
 'required' => string '0' (length=1)
 'scope' => string 'global' (length=6)
1 =>
array
 'attribute_id' => string '93' (length=2)
 'code' => string 'news_from_date' (length=14)
 'type' => string 'date' (length=4)
 'required' => string '0' (length=1)
 'scope' => string 'website' (length=7)
2 =>
array
 ...
...
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Set Special Price

[Edit this page on GitHub](#) 

## Introduction

## Catalog

### Catalog Category

### Catalog Category Attributes

### Catalog Product

- catalog\_product.create
- catalog\_product.currentStore
- catalog\_product.delete
- catalog\_product.getSpecialPrice
- catalog\_product.info
- catalog\_product.list
- catalog\_product.listOfAddition...
- catalog\_product.setSpecialPrice**
- catalog\_product.update

### Catalog Product Attribute

### Catalog Product Attribute

## Media

### Catalog Product Attribute Set

### Catalog Product Custom

### Option

### Catalog Product Custom Option

### Value

### Catalog Product Downloadable

### Link

### Catalog Product Link

### Catalog Product Tag

### Catalog Product Tier Price

### Catalog Product Types

### Catalog Inventory

## Module: Mage\_Catalog

Resource: catalog\_product

### Aliases:

- product

### Method:

- catalog\_product.setSpecialPrice (SOAP V1)
- catalogProductSetSpecialPrice (SOAP V2)

Allows you to set the product special price.

### Aliases:

- product.setSpecialPrice

### Note:

Although the API accepts up to four digits of precision for all price arguments, Magento strongly recommends you pass in two digits to reduce inaccuracy in the tax calculation process. (That is, use a price like 12.35 and not 12.3487).

### Arguments:

| Type   | Name         | Description                                            |
|--------|--------------|--------------------------------------------------------|
| string | sessionId    | Session ID                                             |
| string | productId    | Product ID or SKU                                      |
| string | specialPrice | Product special price                                  |
| string | fromDate     | Date starting from which special price will be applied |

**Checkout** ▾**Create Your Own API****Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

|        |                       |                                                                                         |
|--------|-----------------------|-----------------------------------------------------------------------------------------|
| string | toDate                | Date till which special price will be applied                                           |
| string | storeView             | Store view ID or code (optional)                                                        |
| string | productIdentifierType | Defines whether the product ID or SKU is passed in the 'productId' parameter (optional) |

**Returns:**

| Type        | Name   | Description                                          |
|-------------|--------|------------------------------------------------------|
| boolean/int | result | True (1) if the special price is set for the product |

## Examples

**Request Example SOAP V1**

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product.setSpecialPrice', array('product' => 'testproduct', 'specialPrice' => '77.5', 'fromDate' => '2012-03-29 12:30:51', 'toDate' => '2012-04-29 12:30:51'));
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

**Request Example SOAP V2**

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductSetSpecialPrice($sessionId, '2', '77.5', '2012-03-29 12:30:51', '2012-04-29 12:30:51');
var_dump($result);
```

**Request Example SOAP V2 (WS-I Compliance Mode)**

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductSetSpecialPrice((object)array('sessionId' => $sessionId->result, 'productId' => '2', 'specialPrice' => '77.5', 'fromDate' => '2012-03-29 12:30:51', 'toDate' => '2012-04-29 12:30:51'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Update

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▲

- [catalog\\_product.create](#)
- [catalog\\_product.currentStore](#)
- [catalog\\_product.delete](#)
- [catalog\\_product.getSpecialPrice](#)
- [catalog\\_product.info](#)
- [catalog\\_product.list](#)
- [catalog\\_product.listOfAddition...](#)
- [catalog\\_product.setSpecialPrice](#)
- [catalog\\_product.update](#)

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

## Module: Mage\_Catalog

Resource: catalog\_product

### Aliases:

- product

### Method:

- catalog\_product.update (SOAP V1)
- catalogProductUpdate (SOAP V2)

Allows you to update the required product. Note that you should specify only those parameters which you want to be updated.

### Aliases:

- product.update

### Note:

Although the API accepts up to four digits of precision for all price arguments, Magento strongly recommends you pass in two digits to reduce inaccuracy in the tax calculation process. (That is, use a price like 12.35 and not 12.3487).

### Arguments:

| Type   | Name              | Description                         |
|--------|-------------------|-------------------------------------|
| string | sessionId         | Session ID                          |
| string | product\productId | Product ID                          |
| array  | productData       | Array of catalogProductCreateEntity |
| string | storeView         | Store view ID or code (optional)    |

**Checkout** ▾**Create Your Own API**

|        |                |                                                                            |
|--------|----------------|----------------------------------------------------------------------------|
| string | identifierType | Defines whether the product ID or SKU is passed in the 'product' parameter |
|--------|----------------|----------------------------------------------------------------------------|

**Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance****Returns:**

| Type    | Description                    |
|---------|--------------------------------|
| boolean | True if the product is updated |

The **catalogProductCreateEntity** content is as follows:

| Type          | Name                   | Description                                                               |
|---------------|------------------------|---------------------------------------------------------------------------|
| ArrayOfString | categories             | Array of categories                                                       |
| ArrayOfString | websites               | Array of websites                                                         |
| string        | name                   | Product name                                                              |
| string        | description            | Product description                                                       |
| string        | short_description      | Product short description                                                 |
| string        | weight                 | Product weight                                                            |
| string        | status                 | Product status                                                            |
| string        | url_key                | A relative URL path that can be entered in place of the target path       |
| string        | url_path               | URL path                                                                  |
| string        | visibility             | Product visibility on the frontend                                        |
| ArrayOfString | category_ids           | Array of category IDs                                                     |
| ArrayOfString | website_ids            | Array of website IDs                                                      |
| string        | has_options            | Defines whether the product has options                                   |
| string        | gift_message_available | Defines whether the gift message is available for the product             |
| string        | price                  | Product price                                                             |
| string        | special_price          | Product special price                                                     |
| string        | special_from_date      | Date starting from which the special price will be applied to the product |
| string        | special_to_date        | Date till which the special price will be applied to the product          |
| string        | tax_class_id           | Tax class ID                                                              |
| array         | tier_price             | Array of catalogProductTierPriceEntity                                    |
| string        | meta_title             | Meta title                                                                |
| string        | meta_keyword           | Meta keyword                                                              |
| string        | meta_description       | Meta description                                                          |
| string        | custom_design          | Custom design                                                             |
| string        | custom_layout_update   | Custom layout update                                                      |
| string        | options_container      | Options container                                                         |

|                  |                       |                                                   |
|------------------|-----------------------|---------------------------------------------------|
| associativeArray | additional_attributes | Array of catalogProductAdditionalAttributesEntity |
| array            | stock_data            | Array of catalogInventoryStockItemUpdateEntity    |

**Notes:** The "websites" and "website\_ids" or "categories" and "category\_ids" parameters are interchangeable. In other words, you can specify an array of website IDs (int) and then you don't need to specify the array of website codes (string) and vice versa.

The **catalogProductTierPriceEntity** content is as follows:

| Type   | Name              | Description                                          |
|--------|-------------------|------------------------------------------------------|
| string | customer_group_id | Customer group ID                                    |
| string | website           | Website                                              |
| int    | qty               | Quantity of items to which the price will be applied |
| double | price             | Price that each item will cost                       |

The **catalogInventoryStockItemUpdateEntity** content is as follows:

| Type   | Name                        | Description                                                                    |
|--------|-----------------------------|--------------------------------------------------------------------------------|
| string | qty                         | Quantity of items                                                              |
| int    | is_in_stock                 | Defines whether the item is in stock                                           |
| int    | manage_stock                | Manage stock                                                                   |
| int    | use_config_manage_stock     | Use config manage stock                                                        |
| int    | min_qty                     | Minimum quantity for items to be in stock                                      |
| int    | use_config_min_qty          | Use config settings flag (value defined in the Inventory System Configuration) |
| int    | min_sale_qty                | Minimum quantity allowed in the shopping cart                                  |
| int    | use_config_min_sale_qty     | Use config settings flag                                                       |
| int    | max_sale_qty                | Maximum quantity allowed in the shopping cart                                  |
| int    | use_config_max_sale_qty     | Use config settings flag                                                       |
| int    | is_qty_decimal              | Defines whether the quantity is decimal                                        |
| int    | backorders                  | Backorders status                                                              |
| int    | use_config_backorders       | Use config settings flag (for backorders)                                      |
| int    | notify_stock_qty            | Stock quantity below which a notification will appear                          |
| int    | use_config_notify_stock_qty | Use config settings flag (for stock quantity)                                  |

The **catalogProductAdditionalAttributesEntity** content is as follows:

| Type                  | Name        |
|-----------------------|-------------|
| associativeMultiArray | multi_data  |
| associativeArray      | single_data |

Single Data: array of attributes with only single value

Multi Data: array of attributes which could contain several values

### Faults:

| Fault Code | Fault Message                                 |
|------------|-----------------------------------------------|
| 100        | Requested store view not found.               |
| 101        | Product not exists.                           |
| 102        | Invalid data given. Details in error message. |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->call($session, 'catalog_product.update', array('product_sku',
array(
 'categories' => array(2),
 'websites' => array(1),
 'name' => 'Product name new 2',
 'description' => 'Product description',
 'short_description' => 'Product short description',
 'weight' => '10',
 'status' => '1',
 'url_key' => 'product-url-key',
 'url_path' => 'product-url-path',
 'visibility' => '4',
 'price' => '100',
 'tax_class_id' => 1,
 'meta_title' => 'Product meta title',
 'meta_keyword' => 'Product meta keyword',
 'meta_description' => 'Product meta description'
)));
var_dump ($result);
```

### Request Example SOAP V2

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->catalogProductUpdate($session, 'product_sku', array(
 'categories' => array(2),
 'websites' => array(1),
 'name' => 'Product name new',
 'description' => 'Product description',
 'short_description' => 'Product short description',
 'weight' => '10',
 'status' => '1',
 'url_key' => 'product-url-key',
 'url_path' => 'product-url-path',
 'visibility' => '4',
 'price' => '100',
```

```
'tax_class_id' => 1,
'meta_title' => 'Product meta title',
'meta_keyword' => 'Product meta keyword',
'meta_description' => 'Product meta description'
));

var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductUpdate((object)array('sessionId' => $sessionId-
>result, 'productId' => '1',
'productData' => ((object)array(
'name' => 'Product name updated',
'status' => '1',
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product Attribute

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▼

#### Catalog Product ▼

##### Catalog Product Attribute ^

- product\_attribute.addOption
- product\_attribute.create
- product\_attribute.currentStore
- product\_attribute.info
- product\_attribute.list
- product\_attribute.options
- product\_attribute.remove
- product\_attribute.removeOption
- product\_attribute.types
- product\_attribute.update

##### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

#### Catalog Inventory ▼

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Product Attributes

Allows you to retrieve product attributes and options.

**Resource Name:** catalog\_product\_attribute

#### Aliases:

- product\_attribute

#### Methods:

- [product\\_attribute.currentStore](#) - Set/Get the current store view
- [product\\_attribute.list](#) - Retrieve the attribute list
- [product\\_attribute.options](#) - Retrieve the attribute options
- [product\\_attribute.addOption](#) - Add a new option for attributes with selectable fields
- [product\\_attribute.create](#) - Create a new attribute
- [product\\_attribute.info](#) - Get full information about an attribute with the list of options
- [product\\_attribute.remove](#) - Remove the required attribute
- [product\\_attribute.removeOption](#) - Remove an option for attributes with selectable fields
- [product\\_attribute.types](#) - Get the list of possible attribute types
- [product\\_attribute.update](#) - Update the required attribute

#### Faults:

| Fault Code | Fault Message                   |
|------------|---------------------------------|
| 100        | Requested store view not found. |

|                                    |   |
|------------------------------------|---|
| <b>Checkout</b>                    | ✓ |
| <b>Create Your Own API</b>         |   |
| <b>Customer</b>                    | ✓ |
| <b>Directory</b>                   | ✓ |
| <b>Sales</b>                       | ✓ |
| <b>Enterprise Customer Balance</b> | ✓ |
| <b>Enterprise Gift Card</b>        | ✓ |
| <b>Enterprise Gift Message</b>     | ✓ |
| <b>Miscellaneous</b>               | ✓ |
| <b>WS-I Compliance</b>             |   |

---

|     |                                                                                                                                              |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------|
| 101 | Requested attribute not found.                                                                                                               |
| 102 | Invalid request parameters.                                                                                                                  |
| 103 | Attribute code is invalid. Please use only letters (a-z), numbers (0-9) or underscore (_) in this field, first character should be a letter. |
| 104 | Incorrect attribute type.                                                                                                                    |
| 105 | Unable to save attribute.                                                                                                                    |
| 106 | This attribute cannot be deleted.                                                                                                            |
| 107 | This attribute cannot be edited.                                                                                                             |
| 108 | Unable to add option.                                                                                                                        |
| 109 | Unable to remove option.                                                                                                                     |

### Example:

```

<pre>
<?php
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

echo "<pre>";
// Create new attribute
$attributeToCreate = array(
 "attribute_code" => "new_attribute",
 "scope" => "store",
 "frontend_input" => "select",
 "is_unique" => 0,
 "is_required" => 0,
 "is_configurable" => 0,
 "is_searchable" => 0,
 "is_visible_in_advanced_search" => 0,
 "used_in_product_listing" => 0,
 "additional_fields" => array(
 "is_filterable" => 1,
 "is_filterable_in_search" => 1,
 "position" => 1,
 "used_for_sort_by" => 1
),
 "frontend_label" => array(
 array(
 "store_id" => 0,
 "label" => "A new attribute"
)
)
);

$attributeId = $proxy->call(
 $sessionId,
 "product_attribute.create",
 array(
 $attributeToCreate
)
);

// Update attribute
$attributeToUpdate = array(
 "scope" => "global",

```

```

 "is_unique" => 1,
 "is_required" => 1,
 "is_configurable" => 1,
 "is_searchable" => 1,
 "is_visible_in_advanced_search" => 0,
 "used_in_product_listing" => 0,
 "additional_fields" => array(
 "is_filterable" => 01,
 "is_filterable_in_search" => 0,
 "position" => 2,
 "used_for_sort_by" => 0
),
 "frontend_label" => array(
 array(
 "store_id" => 0,
 "label" => "A Test Attribute"
)
)
);
$proxy->call(
 $sessionId,
 "product_attribute.update",
 array(
 "new_attribute",
 $attributeToUpdate
)
);
// Add option
$optionToAdd = array(
 "label" => array(
 array(
 "store_id" => 0,
 "value" => "New Option"
)
),
 "order" => 0,
 "is_default" => 0
);
$proxy->call(
 $sessionId,
 "product_attribute.addOption",
 array(
 $attributeId,
 $optionToAdd
)
);
// Get info
$resultInfo = $proxy->call(
 $sessionId,
 "product_attribute.info",
 array(
 $attributeId
)
);
echo "info result:\n";
print_r($resultInfo);

// List options
$resultListOptions = $proxy->call(
 $sessionId,

```

```
"product_attribute.options",
array(
 $attributeId
)
);
echo "\n options result:\n";
print_r($resultListOptions);

// Remove option
$result = $proxy->call(
 $sessionId,
 "product_attribute.removeOption",
 array(
 $attributeId,
 $resultInfo['options'][0]['value']
)
);

// remove attribute
$result = $proxy->call(
 $sessionId,
 "product_attribute.remove",
 array(
 $attributeId
)
);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Add Option

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▼

#### Catalog Product ▼

#### Catalog Product Attribute ^

[product\\_attribute.addOption](#)

[product\\_attribute.create](#)

[product\\_attribute.currentStore](#)

[product\\_attribute.info](#)

[product\\_attribute.list](#)

[product\\_attribute.options](#)

[product\\_attribute.remove](#)

[product\\_attribute.removeOption](#)

[product\\_attribute.types](#)

[product\\_attribute.update](#)

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

#### Catalog Inventory ▼

## Module: Product Attributes API

### Resource: product\_attribute

#### Method:

- [product\\_attribute.addOption \(SOAP V1\)](#)
- [catalogProductAttributeAddOption \(SOAP V2\)](#)

Allows you to add a new option for attributes with selectable fields.

#### Arguments:

| Type   | Name      | Description                                       |
|--------|-----------|---------------------------------------------------|
| string | sessionId | Session ID                                        |
| string | attribute | Attribute code or ID                              |
| array  | data      | Array of catalogProductAttributeOptionEntityToAdd |

#### Return:

| Type    | Name   | Description     |
|---------|--------|-----------------|
| boolean | result | True on success |

The **catalogProductAttributeOptionEntityToAdd** content is as follows:

| Type  | Name       | Description                                 |
|-------|------------|---------------------------------------------|
| array | label      | Array of catalogProductAttributeOptionLabel |
| int   | order      | Option order                                |
| int   | is_default | Defines whether the option is a default one |

**Checkout** ▾**Create Your Own API****Customer** ▾**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

The **catalogProductAttributeOptionLabel** content is as follows:

| Type          | Name     | Description             |
|---------------|----------|-------------------------|
| ArrayOfString | store_id | Array of store view IDs |
| string        | value    | Text label              |

**Faults:**

| Fault Code | Fault Message                  |
|------------|--------------------------------|
| 101        | Requested attribute not found. |
| 104        | Incorrect attribute type.      |
| 108        | Unable to add option.          |

## Examples

**Request Example SOAP V1**

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$attributeCode = "new_attribute";
$optionToAdd = array(
 "label" => array(
 array(
 "store_id" => 0,
 "value" => "New Option"
)
),
 "order" => 0,
 "is_default" => 0
);

$result = $proxy->call(
 $sessionId,
 "product_attribute.addOption",
 array(
 $attributeCode,
 $optionToAdd
)
);
```

**Request Example SOAP V2**

```
<?php
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

//V2
$session = $client->login('apiUser', 'apiKey');

// V2 WS-I Mode
// $response = $client->login(array('username' => 'apiUser', 'apiKey' => 'apiKey'));
// $session = $response->result;

$attributeCode = "new_attribute";
```

```

//v2

$label = array (
 array(
 "store_id" => array("0"),
 "value" => "some random data"
)
);

$data = array(
 "label" => $label,
 "order" => "10",
 "is_default" => "1"
);

$orders = $client->catalogProductAttributeAddOption($session, $attributeCode, $data);

//V2 WSI
//WSDL WSI does not describe this method
//$/result = $client->catalogProductAttributeAddOption(...);
//$/orders = $result->result->complexObjectArray;

var_dump ($orders);
?>

```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Create Attribute

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

### Catalog Category ▼

### Catalog Category Attributes ▼

### Catalog Product ▼

### Catalog Product Attribute ^

product\_attribute.addOption

[product\\_attribute.create](#)

product\_attribute.currentStore

product\_attribute.info

product\_attribute.list

product\_attribute.options

product\_attribute.remove

product\_attribute.removeOption

product\_attribute.types

product\_attribute.update

### Catalog Product Attribute

### Media ▼

### Catalog Product Attribute Set ▼

### Catalog Product Custom

### Option ▼

### Catalog Product Custom Option

### Value ▼

### Catalog Product Downloadable

### Link ▼

### Catalog Product Link ▼

### Catalog Product Tag ▼

### Catalog Product Tier Price ▼

### Catalog Product Types ▼

## Catalog Inventory ▼

## Module: Product Attributes API

### Resource: product\_attribute

#### Method:

- `product_attribute.create` (SOAP V1)
- `catalogProductAttributeCreate` (SOAP V2)

Allows you to create a new product attribute.

#### Arguments:

| Type   | Name      | Description                                    |
|--------|-----------|------------------------------------------------|
| string | sessionId | Session ID                                     |
| array  | data      | Array of catalogProductAttributeEntityToCreate |

#### Returns:

| Type | Name   | Description                 |
|------|--------|-----------------------------|
| int  | result | ID of the created attribute |

The `catalogProductAttributeEntityToCreate` content is as follows:

| Type   | Name           | Description                                                                      |
|--------|----------------|----------------------------------------------------------------------------------|
| string | attribute_code | Attribute code                                                                   |
| string | frontend_input | Attribute type                                                                   |
| string | scope          | Attribute scope. Possible values are as follows: 'store', 'website', or 'global' |

|                                      |                  |                               |                                                                                                                                                                    |
|--------------------------------------|------------------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Checkout</b> ↴                    | string           | default_value                 | Attribute default value                                                                                                                                            |
| <b>Create Your Own API</b>           | int              | is_unique                     | Defines whether the attribute is unique                                                                                                                            |
| <b>Customer</b> ↴                    | int              | is_required                   | Defines whether the attribute is required                                                                                                                          |
| <b>Directory</b> ↴                   |                  |                               |                                                                                                                                                                    |
| <b>Sales</b> ↴                       | ArrayOfString    | apply_to                      | Apply to. Empty for "Apply to all" or array of the following possible values: 'simple', 'grouped', 'configurable', 'virtual', 'bundle', 'downloadable', 'giftcard' |
| <b>Enterprise Customer Balance</b> ↴ | int              | is_configurable               | Defines whether the attribute can be used for configurable products                                                                                                |
| <b>Enterprise Gift Card</b> ↴        | int              | is_searchable                 | Defines whether the attribute can be used in Quick Search                                                                                                          |
| <b>Enterprise Gift Message</b> ↴     | int              | is_visible_in_advanced_search | Defines whether the attribute can be used in Advanced Search                                                                                                       |
| <b>Miscellaneous</b> ↴               | int              | is_comparable                 | Defines whether the attribute can be compared on the frontend                                                                                                      |
| <b>WS-I Compliance</b>               | int              | is_used_for_promo_rules       | Defines whether the attribute can be used for promo rules                                                                                                          |
|                                      | int              | is_visible_on_front           | Defines whether the attribute is visible on the frontend                                                                                                           |
|                                      | int              | used_in_product_listing       | Defines whether the attribute can be used in product listing                                                                                                       |
|                                      | associativeArray | additional_fields             | Array of additional fields                                                                                                                                         |
|                                      | array            | frontend_label                | Array of catalogProductAttributeFrontendLabel                                                                                                                      |

The **catalogProductAttributeFrontendLabelEntity** content is as follows:

| Type   | Name     | Description |
|--------|----------|-------------|
| string | store_id | Store ID    |
| string | label    | Text label  |

**Notes:** The "label" value for the "store\_id" value set to 0 must be specified. An attribute cannot be created without specifying the label for store\_id=0.

The **AdditionaFieldsEntity** array of additional fields for the **text** type is as follows:

| Type    | Name                     | Description                                                                                                                                                                                                                                                        |
|---------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| string  | frontend_class           | Input Validation for Store Owner. Possible values are as follows: 'validate-number' (Decimal Number), 'validate-digits' (Integer Number), 'validate-email', 'validate-url', 'validate-alpha' (Letters), 'validate-alphanum' (Letters (a-z, A-Z), or Numbers (0-9)) |
| boolean | is_html_allowed_on_front | Defines whether the HTML tags are allowed on the frontend                                                                                                                                                                                                          |
| boolean | used_for_sort_by         | Defines whether it is used for sorting in product listing                                                                                                                                                                                                          |

The **AdditionaFieldsEntity** array of additional fields for the **text area** type is as follows:

| Type    | Name                     | Description                                               |
|---------|--------------------------|-----------------------------------------------------------|
| boolean | is_wysiwyg_enabled       | Enable WYSIWYG flag                                       |
| boolean | is_html_allowed_on_front | Defines whether the HTML tags are allowed on the frontend |

The **AdditionaFieldsEntity** array of additional fields for the **date** and **boolean** types is as follows:

| Type | Name | Description |
|------|------|-------------|
|      |      |             |

|         |                  |                                                           |
|---------|------------------|-----------------------------------------------------------|
| boolean | used_for_sort_by | Defines whether it is used for sorting in product listing |
|---------|------------------|-----------------------------------------------------------|

The **AdditionaFieldsEntity** array of additional fields for the **multiselect** type is as follows:

| Type    | Name                    | Description                                                     |
|---------|-------------------------|-----------------------------------------------------------------|
| boolean | is_filterable           | Defines whether it is used in layered navigation                |
| boolean | is_filterable_in_search | Defines whether it is used in search results layered navigation |
| int     | position                | Position                                                        |

The **AdditionaFieldsEntity** array of additional fields for the **select** and **price** types is as follows:

| Type    | Name                    | Description                                                     |
|---------|-------------------------|-----------------------------------------------------------------|
| boolean | is_filterable           | Defines whether it is used in layered navigation                |
| boolean | is_filterable_in_search | Defines whether it is used in search results layered navigation |
| int     | position                | Position                                                        |
| boolean | used_for_sort_by        | Defines whether it is used for sorting in product listing       |

### Faults:

| Fault Code | Fault Message                                                                                                                                |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 102        | Invalid request parameters.                                                                                                                  |
| 103        | Attribute code is invalid. Please use only letters (a-z), numbers (0-9) or underscore (_) in this field, first character should be a letter. |
| 104        | Incorrect attribute type.                                                                                                                    |
| 105        | Unable to save attribute.                                                                                                                    |

## Examples

### Request Example SOAP V1

```
<?php

$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$attributeToUpdate = array(
 "scope" => "global",
 "default_value" => "100",
 "frontend_input" => "text",
 "is_unique" => 0,
 "is_required" => 0,
 "is_configurable" => 0,
 "is_searchable" => 0,
 "is_visible_in_advanced_search" => 0,
 "used_in_product_listing" => 0,
 "additional_fields" => array(
 "is_filterable" => 1,
 "is_filterable_in_search" => 1,
 "position" => 100
)
);
```

```

 "is_filterable" => 1,
 "is_filterable_in_search" => 1,
 "position" => 1,
 "used_for_sort_by" => 1
),
 "frontend_label" => array(
 array(
 "store_id" => 0,
 "label" => "Updated attribute"
)
)
);

$attributeCode = 'code1';

$result = $client->call($session, 'product_attribute.update', array($attributeCode,
$attributeToUpdate));
var_dump ($result);

// If you don't need the session anymore
// $client->endSession($session);

?>

```

## Request Example SOAP V2

```

<?php
//ini_set("soap.wsdl_cache_enabled", 0);

$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

//V2
$session = $client->login('apiUser', 'apiKey');

// V2 WS-I Mode
//$response = $client->login(array('username' => 'apiUser', 'apiKey' => 'apiKey'));
// $session = $response->result;

//v2

$data = array(
 "attribute_code" => "test_attribute",
 "frontend_input" => "text",
 "scope" => "1",
 "default_value" => "1",
 "is_unique" => 0,
 "is_required" => 0,
 "apply_to" => array("simple"),
 "is_configurable" => 0,
 "is_searchable" => 0,
 "is_visible_in_advanced_search" => 0,
 "is_comparable" => 0,
 "is_used_for_promo_rules" => 0,
 "is_visible_on_front" => 0,
 "used_in_product_listing" => 0,
 "additional_fields" => array(),
 "frontend_label" => array(array("store_id" => "0", "label" => "some label"))
);

$orders = $client->catalogProductAttributeCreate($session, $data);

```

```
//V2 WSI
//WSDL WSI Sample is not complete
//$result = $client->catalogProductAttributeCreate(array("sessionId" => $session,
//"data" => $data));
//$/orders = $result->result->complexObjectArray;

echo 'Number of results: ' . count($orders) . '
';
var_dump ($orders);
?>
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Current Store

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▼

#### Catalog Product ▼

#### Catalog Product Attribute ▼

product\_attribute.addOption

product\_attribute.create

[product\\_attribute.currentStore](#)

product\_attribute.info

product\_attribute.list

product\_attribute.options

product\_attribute.remove

product\_attribute.removeOption

product\_attribute.types

product\_attribute.update

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

#### Catalog Inventory ▼

## Module: Mage\_Catalog

Resource: catalog\_product\_attribute

### Aliases:

- product\_attribute

### Method:

- catalog\_product\_attribute.currentStore (SOAP V1)
- catalogProductAttributeCurrentStore (SOAP V2)

Allows you to set/get the current store view.

### Aliases:

- product\_attribute.currentStore

### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | storeView | Store view ID or code (optional) |

### Returns:

| Type | Name      | Description   |
|------|-----------|---------------|
| int  | storeView | Store view ID |

## Examples

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_attribute.currentStore',
'english');
var_dump ($result);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeCurrentStore($sessionId, 'english');
var_dump($result);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▼

#### Catalog Product ▼

#### Catalog Product Attribute ^

product\_attribute.addOption

product\_attribute.create

product\_attribute.currentStore

[product\\_attribute.info](#)

product\_attribute.list

product\_attribute.options

product\_attribute.remove

product\_attribute.removeOption

product\_attribute.types

product\_attribute.update

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

#### Catalog Inventory ▼

## Module: Product Attributes API

### Resource: product\_attribute

#### Method:

- `product_attribute.info` (SOAP V1)
- `catalogProductAttributeInfo` (SOAP V2)

Allows you to get full information about a required attribute with the list of options.

#### Arguments:

| Type   | Name      | Description          |
|--------|-----------|----------------------|
| string | sessionId | Session ID           |
| string | attribute | Attribute code or ID |

#### Return:

| Type  | Name   | Description                            |
|-------|--------|----------------------------------------|
| array | result | Array of catalogProductAttributeEntity |

The `catalogProductAttributeEntity` content is as follows:

| Type   | Name           | Description     |
|--------|----------------|-----------------|
| string | attribute_id   | Attribute ID    |
| string | attribute_code | Attribute code  |
| string | frontend_input | Attribute type  |
| string | scope          | Attribute scope |

|                                    |   |
|------------------------------------|---|
| <b>Checkout</b>                    | ✓ |
| <b>Create Your Own API</b>         |   |
| <b>Customer</b>                    | ✓ |
| <b>Directory</b>                   | ✓ |
| <b>Sales</b>                       | ✓ |
| <b>Enterprise Customer Balance</b> | ✓ |
| <b>Enterprise Gift Card</b>        | ✓ |
| <b>Enterprise Gift Message</b>     | ✓ |
| <b>Miscellaneous</b>               | ✓ |
| <b>WS-I Compliance</b>             |   |

|                  |                               |                                                                                                                                                                    |
|------------------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| string           | default_value                 | Attribute default value                                                                                                                                            |
| int              | is_unique                     | Defines whether the attribute is unique                                                                                                                            |
| int              | is_required                   | Defines whether the attribute is required                                                                                                                          |
| ArrayOfString    | apply_to                      | Apply to. Empty for "Apply to all" or array of the following possible values: 'simple', 'grouped', 'configurable', 'virtual', 'bundle', 'downloadable', 'giftcard' |
| int              | is_configurable               | Defines whether the attribute can be used for configurable products                                                                                                |
| int              | is_searchable                 | Defines whether the attribute can be used in Quick Search                                                                                                          |
| int              | is_visible_in_advanced_search | Defines whether the attribute can be used in Advanced Search                                                                                                       |
| int              | is_comparable                 | Defines whether the attribute can be compared on the frontend                                                                                                      |
| int              | is_used_for_promo_rules       | Defines whether the attribute can be used for promo rules                                                                                                          |
| int              | is_visible_on_front           | Defines whether the attribute is visible on the frontend                                                                                                           |
| int              | used_in_product_listing       | Defines whether the attribute can be used in product listing                                                                                                       |
| associativeArray | additional_fields             | Array of additional fields                                                                                                                                         |
| array            | options                       | Array of catalogAttributeOptionEntity                                                                                                                              |
| array            | frontend_label                | Array of catalogProductAttributeFrontendLabel                                                                                                                      |

The **catalogAttributeOptionEntity** content is as follows:

| Type   | Name  | Description |
|--------|-------|-------------|
| string | label | Text label  |
| string | value | Option ID   |

The **catalogProductAttributeFrontendLabelEntity** content is as follows:

| Type   | Name     | Description |
|--------|----------|-------------|
| string | store_id | Store ID    |
| string | label    | Text label  |

The **AdditionaFieldsEntity** array of additional fields for the **text** type is as follows:

| Type    | Name                     | Description                                                                                                                                                                                                                                         |
|---------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| string  | frontend_class           | Input Validation for Store Owner. Possible values: 'validate-number' (Decimal Number), 'validate-digits' (Integer Number), 'validate-email', 'validate-url', 'validate-alpha' (Letters), 'validate-alphanum' (Letters (a-z, A-Z), or Numbers (0-9)) |
| boolean | is_html_allowed_on_front | Defines whether the HTML tags are allowed on the frontend                                                                                                                                                                                           |
| boolean | used_for_sort_by         | Defines whether it is used for sorting in product listing                                                                                                                                                                                           |

The **AdditionaFieldsEntity** array of additional fields for the **text area** type is as follows:

| Type | Name | Description |
|------|------|-------------|
|      |      |             |

|         |                          |                                                           |
|---------|--------------------------|-----------------------------------------------------------|
| boolean | is_wysiwyg_enabled       | Enable WYSIWYG flag                                       |
| boolean | is_html_allowed_on_front | Defines whether the HTML tags are allowed on the frontend |

The **AdditionaFieldsEntity** array of additional fields for the **date** and **boolean** types is as follows:

| Type    | Name             | Description                                               |
|---------|------------------|-----------------------------------------------------------|
| boolean | used_for_sort_by | Defines whether it is used for sorting in product listing |

The **AdditionaFieldsEntity** array of additional fields for the **multiselect** type is as follows:

| Type    | Name                    | Description                                                     |
|---------|-------------------------|-----------------------------------------------------------------|
| boolean | is_filterable           | Defines whether it used in layered navigation                   |
| boolean | is_filterable_in_search | Defines whether it is used in search results layered navigation |
| int     | position                | Position                                                        |

The **AdditionaFieldsEntity** array of additional fields for the **select** and **price** types is as follows:

| Type    | Name                    | Description                                                     |
|---------|-------------------------|-----------------------------------------------------------------|
| boolean | is_filterable           | Defines whether it used in layered navigation                   |
| boolean | is_filterable_in_search | Defines whether it is used in search results layered navigation |
| int     | position                | Position                                                        |
| boolean | used_for_sort_by        | Defines whether it is used for sorting in product listing       |

### Faults:

| Fault Code | Fault Message                  |
|------------|--------------------------------|
| 101        | Requested attribute not found. |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'product_attribute.info', '11');
var_dump ($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary
```

```
$result = $proxy->catalogProductAttributeInfo($sessionId, '11');
var_dump($result);
```

## Response Example SOAP V1

```
array
 'attribute_id' => string '11' (length=3)
 'attribute_code' => string 'new_special_price' (length=17)
 'frontend_input' => string 'text' (length=4)
 'default_value' => null
 'is_unique' => string '0' (length=1)
 'is_required' => string '0' (length=1)
 'apply_to' =>
 array
 empty
 'is_configurable' => string '0' (length=1)
 'is_searchable' => string '0' (length=1)
 'is_visible_in_advanced_search' => string '0' (length=1)
 'is_comparable' => string '0' (length=1)
 'is_used_for_promo_rules' => string '0' (length=1)
 'is_visible_on_front' => string '0' (length=1)
 'used_in_product_listing' => string '0' (length=1)
 'frontend_label' =>
 array
 0 =>
 array
 'store_id' => int 0
 'label' => string 'special price' (length=13)
 1 =>
 array
 'store_id' => int 2
 'label' => string 'special price' (length=13)
 'scope' => string 'store' (length=5)
 'additional_fields' =>
 array
 'frontend_class' => null
 'is_html_allowed_on_front' => string '1' (length=1)
 'used_for_sort_by' => string '0' (length=1)
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▼

#### Catalog Product ▼

#### Catalog Product Attribute ▼

- product\_attribute.addOption
- product\_attribute.create
- product\_attribute.currentStore
- product\_attribute.info
- product\_attribute.list**
- product\_attribute.options
- product\_attribute.remove
- product\_attribute.removeOption
- product\_attribute.types
- product\_attribute.update

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

## Module: Mage\_Catalog

Resource: catalog\_product\_attribute

#### Aliases:

- product\_attribute

#### Method:

- catalog\_product\_attribute.list (SOAP V1)
- catalogProductAttributeList (SOAP V2)

Allows you to retrieve the list of product attributes.

#### Aliases:

- product\_attribute.list

#### Arguments:

| Type   | Name      | Description             |
|--------|-----------|-------------------------|
| string | sessionId | Session ID              |
| int    | setId     | ID of the attribute set |

#### Returns:

| Type  | Name   | Description                     |
|-------|--------|---------------------------------|
| array | result | Array of catalogAttributeEntity |

The **catalogAttributeEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|                                             |   |
|---------------------------------------------|---|
| <a href="#">Checkout</a>                    | ✓ |
| <a href="#">Create Your Own API</a>         |   |
| <a href="#">Customer</a>                    | ✓ |
| <a href="#">Directory</a>                   | ✓ |
| <a href="#">Sales</a>                       | ✓ |
| <a href="#">Enterprise Customer Balance</a> | ✓ |
| <a href="#">Enterprise Gift Card</a>        | ✓ |
| <a href="#">Enterprise Gift Message</a>     | ✓ |
| <a href="#">Miscellaneous</a>               | ✓ |
| <a href="#">WS-I Compliance</a>             |   |

|        |              |                                                                   |
|--------|--------------|-------------------------------------------------------------------|
| int    | attribute_id | Attribute ID                                                      |
| string | code         | Attribute code                                                    |
| string | type         | Attribute type                                                    |
| string | required     | Defines whether the attribute is required                         |
| string | scope        | Attribute scope. Possible values: 'store', 'website', or 'global' |

#### Faults:

*No faults*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$setId = 4; // Existing attribute set id

$result = $proxy->call(
 $sessionId,
 "product_attribute.list",
 array(
 $setId
)
);
echo "<pre>";
var_dump($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeList($sessionId, '4');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductAttributeList((object)array('sessionId' => $sessionId->result, 'setId' => '4'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
 array
```

```
'attribute_id' => string '71' (length=2)
'code' => string 'name' (length=4)
'type' => string 'text' (length=4)
'required' => string '1' (length=1)
'scope' => string 'store' (length=5)
1 =>
array
'attribute_id' => string '72' (length=2)
'code' => string 'description' (length=11)
'type' => string 'textarea' (length=8)
'required' => string '1' (length=1)
'scope' => string 'store' (length=5)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Attribute Options

[Edit this page on GitHub](#) 

## Introduction

## Catalog

### Catalog Category

### Catalog Category Attributes

### Catalog Product

### Catalog Product Attribute

- product\_attribute.addOption
- product\_attribute.create
- product\_attribute.currentStore
- product\_attribute.info
- product\_attribute.list
- product\_attribute.options**
- product\_attribute.remove
- product\_attribute.removeOption
- product\_attribute.types
- product\_attribute.update

### Catalog Product Attribute

## Media

### Catalog Product Attribute Set

### Catalog Product Custom Option

### Catalog Product Custom Option Value

### Catalog Product Downloadable Link

### Catalog Product Link

### Catalog Product Tag

### Catalog Product Tier Price

### Catalog Product Types

## Catalog Inventory

## Module: Mage\_Catalog

Resource: catalog\_product\_attribute

### Aliases:

- product\_attribute

### Method:

- catalog\_product\_attribute.options (SOAP V1)
- catalogProductAttributeOptions (SOAP V2)

Allows you to retrieve the product attribute options.

### Aliases:

- product\_attribute.options

### Arguments:

| Type   | Name        | Description                      |
|--------|-------------|----------------------------------|
| string | sessionId   | Session ID                       |
| string | attributeld | Attribute ID or code             |
| string | storeView   | Store view ID or code (optional) |

### Returns:

| Type  | Name   | Description                           |
|-------|--------|---------------------------------------|
| array | result | Array of catalogAttributeOptionEntity |

The **catalogAttributeOptionEntity** content is as follows:

|                                             |   |
|---------------------------------------------|---|
| <a href="#">Checkout</a>                    | ✓ |
| <a href="#">Create Your Own API</a>         |   |
| <a href="#">Customer</a>                    | ✓ |
| <a href="#">Directory</a>                   | ✓ |
| <a href="#">Sales</a>                       | ✓ |
| <a href="#">Enterprise Customer Balance</a> | ✓ |
| <a href="#">Enterprise Gift Card</a>        | ✓ |
| <a href="#">Enterprise Gift Message</a>     | ✓ |
| <a href="#">Miscellaneous</a>               | ✓ |
| <a href="#">WS-I Compliance</a>             |   |

| Type   | Name  | Description  |
|--------|-------|--------------|
| string | label | Option label |
| string | value | Option value |

#### Faults:

| Fault Code | Fault Message                  |
|------------|--------------------------------|
| 101        | Requested attribute not found. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$attributeId = 11; // Existing selectable attribute ID

$result = $proxy->call(
 $sessionId,
 "product_attribute.options",
 array(
 $attributeId
)
);
echo "<pre>";
var_dump($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeOptions($sessionId, '11');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductAttributeOptions((object)array('sessionId' => $sessionId->result, 'attributeId' => '11'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
 1 =>
 array
 'value' => string '5' (length=1)
 'label' => string 'blue' (length=4)
```

```
2 =>
 array
 'value' => string '4' (length=1)
 'label' => string 'green' (length=5)
3 =>
 array
 'value' => string '3' (length=1)
 'label' => string 'yellow' (length=6)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Attribute Remove

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ^

[product\\_attribute.addOption](#)

[product\\_attribute.create](#)

[product\\_attribute.currentStore](#)

[product\\_attribute.info](#)

[product\\_attribute.list](#)

[product\\_attribute.options](#)

[product\\_attribute.remove](#)

[product\\_attribute.removeOption](#)

[product\\_attribute.types](#)

[product\\_attribute.update](#)

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

## Module: Product Attributes API

### Resource: product\_attribute

#### Method:

- [product\\_attribute.remove \(SOAP V1\)](#)
- [catalogProductAttributeRemove \(SOAP V2\)](#)

Allows you to remove the required attribute from a product.

#### Arguments:

| Type   | Name      | Description          |
|--------|-----------|----------------------|
| string | sessionId | Session ID           |
| string | attribute | Attribute code or ID |

#### Return:

| Type    | Description                      |
|---------|----------------------------------|
| boolean | True if the attribute is removed |

#### Faults:

| Fault Code | Fault Message                     |
|------------|-----------------------------------|
| 101        | Requested attribute not found.    |
| 106        | This attribute cannot be deleted. |

## Examples

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$attributeCode = "11";

$result = $proxy->call(
 $sessionId,
 "product_attribute.remove",
 array(
 $attributeCode
)
);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeRemove($sessionId, '11');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$session = $client->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $client->catalogProductAttributeRemove((object)array('sessionId' => $session->result, 'attribute' => '11'));

var_dump ($result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Remove Option

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▼

#### Catalog Product ▼

#### Catalog Product Attribute ^

product\_attribute.addOption

product\_attribute.create

product\_attribute.currentStore

product\_attribute.info

product\_attribute.list

product\_attribute.options

product\_attribute.remove

**product\_attribute.removeOption**

product\_attribute.types

product\_attribute.update

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

#### Catalog Inventory ▼

## Module: Product Attributes API

### Resource: product\_attribute

#### Method:

- product\_attribute.removeOption (SOAP V1)
- catalogProductAttributeRemoveOption (SOAP V2)

Allows you to remove the option for an attribute.

#### Arguments:

| Type   | Name      | Description          |
|--------|-----------|----------------------|
| string | sessionId | Session ID           |
| string | attribute | Attribute code or ID |
| string | optionId  | Option ID            |

#### Return:

| Type    | Description                   |
|---------|-------------------------------|
| boolean | True if the option is removed |

#### Faults:

| Fault Code | Fault Message                  |
|------------|--------------------------------|
| 101        | Requested attribute not found. |
| 104        | Incorrect attribute type.      |
| 109        | Unable to remove option.       |

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Examples

### Request example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$attributeCode = "2";
$optionId = 11; // Existing option ID

$result = $proxy->call(
 $sessionId,
 "product_attribute.removeOption",
 array(
 $attributeCode,
 $optionId
)
);
```

### Request example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeRemoveOption($sessionId, '2', '11');
var_dump($result);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Attribute Types

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

### Catalog Category ▼

### Catalog Category Attributes ▼

### Catalog Product ▼

### Catalog Product Attribute ^

product\_attribute.addOption  
product\_attribute.create  
product\_attribute.currentStore  
product\_attribute.info  
product\_attribute.list  
product\_attribute.options  
product\_attribute.remove  
product\_attribute.removeOption  
**product\_attribute.types**  
product\_attribute.update

### Catalog Product Attribute

### Media ▼

### Catalog Product Attribute Set ▼

### Catalog Product Custom

### Option ▼

### Catalog Product Custom Option

### Value ▼

### Catalog Product Downloadable

### Link ▼

### Catalog Product Link ▼

### Catalog Product Tag ▼

### Catalog Product Tier Price ▼

### Catalog Product Types ▼

### Catalog Inventory ▼

## Module: Product Attributes API

### Resource: product\_attribute

#### Method:

- `product_attribute.types` (SOAP V1)
- `catalogProductAttributeTypes` (SOAP V2)

Allows you to retrieve the list of possible attribute types.

#### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

#### Return:

| Type  | Name   | Description                           |
|-------|--------|---------------------------------------|
| array | result | Array of catalogAttributeOptionEntity |

The **catalogAttributeOptionEntity** content is as follows:

| Type   | Name  | Description  |
|--------|-------|--------------|
| string | label | Option label |
| string | value | Option value |

#### Faults:

*No Faults.*

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->call(
 $sessionId,
 "product_attribute.types"
);

echo "<pre>";
var_dump($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeTypes($sessionId);
var_dump($result);
```

### Response Example SOAP V1

```
array
 0 =>
 'value' => 'text'
 'label' => 'Text Field'
 1 =>
 'value' => 'textarea'
 'label' => 'Text Area'
 2 =>
 'value' => 'date'
 'label' => 'Date'
 3 =>
 'value' => 'boolean'
 'label' => 'Yes/No'
 4 =>
 'value' => 'multiselect'
 'label' => 'Multiple Select'
 5 =>
 'value' => 'select'
 'label' => 'Dropdown'
 6 =>
 'value' => 'price'
 'label' => 'Price'
 7 =>
 'value' => 'media_image'
 'label' => 'Media Image'
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute Update

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

#### Catalog Category ▼

#### Catalog Category Attributes ▼

#### Catalog Product ▼

#### Catalog Product Attribute ▲

- product\_attribute.addOption
- product\_attribute.create
- product\_attribute.currentStore
- product\_attribute.info
- product\_attribute.list
- product\_attribute.options
- product\_attribute.remove
- product\_attribute.removeOption
- product\_attribute.types
- [product\\_attribute.update](#)

#### Catalog Product Attribute

#### Media ▼

#### Catalog Product Attribute Set ▼

#### Catalog Product Custom

#### Option ▼

#### Catalog Product Custom Option

#### Value ▼

#### Catalog Product Downloadable

#### Link ▼

#### Catalog Product Link ▼

#### Catalog Product Tag ▼

#### Catalog Product Tier Price ▼

#### Catalog Product Types ▼

#### Catalog Inventory ▼

## Module: Product Attributes API

### Resource: product\_attribute

#### Method:

- `product_attribute.update` (SOAP V1)
- `catalogProductAttributeUpdate` (SOAP V2)

Allows you to update the required attribute.

#### Arguments:

| Type   | Name      | Description                                    |
|--------|-----------|------------------------------------------------|
| string | sessionId | Session ID                                     |
| string | attribute | Attribute code or ID                           |
| array  | data      | Array of catalogProductAttributeEntityToUpdate |

#### Returns:

| Type    | Description                      |
|---------|----------------------------------|
| boolean | True if the attribute is updated |

The `catalogProductAttributeEntityToUpdate` content is as follows:

| Type   | Name          | Description                                                                      |
|--------|---------------|----------------------------------------------------------------------------------|
| string | scope         | Attribute scope. Possible values are as follows: 'store', 'website', or 'global' |
| string | default_value | Attribute default value                                                          |

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

|                  |                               |                                                                                                                                                                    |
|------------------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int              | is_unique                     | Defines whether the attribute is unique                                                                                                                            |
| int              | is_required                   | Defines whether the attribute is required                                                                                                                          |
| ArrayOfString    | apply_to                      | Apply to. Empty for "Apply to all" or array of the following possible values: 'simple', 'grouped', 'configurable', 'virtual', 'bundle', 'downloadable', 'giftcard' |
| int              | is_configurable               | Defines whether the attribute can be used for configurable products                                                                                                |
| int              | is_searchable                 | Defines whether the attribute can be used in Quick Search                                                                                                          |
| int              | is_visible_in_advanced_search | Defines whether the attribute can be used in Advanced Search                                                                                                       |
| int              | is_comparable                 | Defines whether the attribute can be compared on the frontend                                                                                                      |
| int              | is_used_for_promo_rules       | Defines whether the attribute can be used for promo rules                                                                                                          |
| int              | is_visible_on_front           | Defines whether the attribute can be visible on the frontend                                                                                                       |
| int              | used_in_product_listing       | Defines whether the attribute can be used in product listing                                                                                                       |
| associativeArray | additional_fields             | Array of additional fields                                                                                                                                         |
| array            | frontend_label                | Array of catalogProductAttributeFrontendLabel                                                                                                                      |

The **AdditionaFieldsEntity** array of additional fields for the **text** type is as follows:

| Type    | Name                     | Description                                                                                                                                                                                                                                         |
|---------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| string  | frontend_class           | Input Validation for Store Owner. Possible values: 'validate-number' (Decimal Number), 'validate-digits' (Integer Number), 'validate-email', 'validate-url', 'validate-alpha' (Letters), 'validate-alphanum' (Letters (a-z, A-Z), or Numbers (0-9)) |
| boolean | is_html_allowed_on_front | Defines whether the HTML tags are allowed on the frontend                                                                                                                                                                                           |
| boolean | used_for_sort_by         | Defines whether it is used for sorting in product listing                                                                                                                                                                                           |

The **AdditionaFieldsEntity** array of additional fields for the **text area** type is as follows:

| Type    | Name                     | Description                                               |
|---------|--------------------------|-----------------------------------------------------------|
| boolean | is_wysiwyg_enabled       | Enable WYSIWYG flag                                       |
| boolean | is_html_allowed_on_front | Defines whether the HTML tags are allowed on the frontend |

The **AdditionaFieldsEntity** array of additional fields for the **date** and **boolean** types is as follows:

| Type    | Name             | Description                                               |
|---------|------------------|-----------------------------------------------------------|
| boolean | used_for_sort_by | Defines whether it is used for sorting in product listing |

The **AdditionaFieldsEntity** array of additional fields for the **multiselect** type is as follows:

| Type    | Name                    | Description                                                     |
|---------|-------------------------|-----------------------------------------------------------------|
| boolean | is_filterable           | Defines whether it used in layered navigation                   |
| boolean | is_filterable_in_search | Defines whether it is used in search results layered navigation |
| integer | position                | Position                                                        |

The **AdditionaFieldsEntity** array of additional fields for the **select** and **price** types is as follows:

| Type    | Name                    | Description                                                     |
|---------|-------------------------|-----------------------------------------------------------------|
| boolean | is_filterable           | Defines whether it is used in layered navigation                |
| boolean | is_filterable_in_search | Defines whether it is used in search results layered navigation |
| integer | position                | Position                                                        |
| boolean | used_for_sort_by        | Defines whether it is used for sorting in product listing       |

The **catalogProductAttributeFrontendLabel** content is as follows:

| Type   | Name     | Description |
|--------|----------|-------------|
| string | store_id | Store ID    |
| string | label    | Text label  |

#### Faults:

| Fault Code | Fault Message                    |
|------------|----------------------------------|
| 101        | Requested attribute not found.   |
| 105        | Unable to save attribute.        |
| 107        | This attribute cannot be edited. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$attributeToUpdate = array(
 "scope" => "global",
 "is_unique" => 1,
 "is_required" => 1,
 "is_configurable" => 1,
 "is_searchable" => 1,
 "is_visible_in_advanced_search" => 0,
 "used_in_product_listing" => 0,
 "additional_fields" => array(
 "is_filterable" => 1,
 "is_filterable_in_search" => 0,
 "position" => 2,
 "used_for_sort_by" => 0
),
 "frontend_label" => array(
 array(
 "store_id" => 0,
 "label" => "A Test Attribute"
)
)
);

$result = $proxy->call(
 $sessionId,
 "product_attribute.update",
 array(
 "attribute" => $attributeToUpdate
)
);
```

```

 array(
 $attributeToUpdate
)
);

```

## Request Example SOAP V2

```

<?php
//ini_set("soap.wsdl_cache_enabled", 0);

$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

//V2
$session = $client->login('apiUser', 'apiKey');

// V2 WS-I Mode
//$/response = $client->login(array('username' => 'apiUser', 'apiKey' => 'apiKey'));
//$/session = $response->result;

//v2
$attributeCode = "code1";
$data = array(
 "frontend_input" => "text",
 "scope" => "1",
 "default_value" => "1",
 "is_unique" => 0,
 "is_required" => 0,
 "apply_to" => array("simple"),
 "is_configurable" => 0,
 "is_searchable" => 0,
 "is_visible_in_advanced_search" => 0,
 "is_comparable" => 0,
 "is_used_for_promo_rules" => 0,
 "is_visible_on_front" => 0,
 "used_in_product_listing" => 0,
 "additional_fields" => array(),
 "frontend_label" => array(array("store_id" => "0", "label" => "some random label
updated")))
);

$orders = $client->catalogProductAttributeUpdate($session, $attributeCode, $data);

//V2 WSI
//WSDL WSI Sample is not complete
//$/result = $client->catalogProductAttributeCreate(array("sessionId" => $session,
//"data" => $data));
//$/orders = $result->result->complexTypeArray;

echo 'Number of results: ' . count($orders) . '
';
var_dump ($orders);
?>

```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product Attribute Media

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▾

[Catalog Category Attributes](#) ▾

[Catalog Product](#) ▾

[Catalog Product Attribute](#) ▾

[Catalog Product Attribute](#)

### Media ^

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[Catalog Product Attribute Set](#) ▾

[Catalog Product Custom](#)

[Option](#) ▾

[Catalog Product Custom Option](#)

[Value](#) ▾

[Catalog Product Downloadable](#)

[Link](#) ▾

[Catalog Product Link](#) ▾

[Catalog Product Tag](#) ▾

[Catalog Product Tier Price](#) ▾

[Catalog Product Types](#) ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Product Images

Allows you to manage product images.

**Resource Name:** catalog\_product\_attribute\_media

#### Aliases:

- product\_attribute\_media
- product\_media

#### Methods:

- [catalog\\_product\\_attribute\\_media.currentStore](#) - Set/Get the current store view
- [catalog\\_product\\_attribute\\_media.list](#) - Retrieve the product images
- [catalog\\_product\\_attribute\\_media.info](#) - Retrieve the specified product image
- [catalog\\_product\\_attribute\\_media.types](#) - Retrieve product image types
- [catalog\\_product\\_attribute\\_media.create](#) - Upload a new image for a product
- [catalog\\_product\\_attribute\\_media.update](#) - Update an image for a product
- [catalog\\_product\\_attribute\\_media.remove](#) - Remove an image for a product

### Faults

| Fault Code | Fault Message                                 |
|------------|-----------------------------------------------|
| 100        | Requested store view not found.               |
| 101        | Product not exists.                           |
| 102        | Invalid data given. Details in error message. |

**Customer** ▾

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

|     |                                                        |
|-----|--------------------------------------------------------|
| 103 | Requested image not exists in product images' gallery. |
| 104 | Image creation failed. Details in error message.       |
| 105 | Image not updated. Details in error message.           |
| 106 | Image not removed. Details in error message.           |
| 107 | Requested product doesn't support images               |

## Examples

### Example 1. Working with product images

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$newImage = array(
 'file' => array(
 'name' => 'file_name',
 'content' => base64_encode(file_get_contents('product.jpg')),
 'mime' => 'image/jpeg'
),
 'label' => 'Cool Image Through Soap',
 'position' => 2,
 'types' => array('small_image'),
 'exclude' => 0
);

$imageFilename = $proxy->call($sessionId, 'product_media.create', array('Sku',
$newImage));

var_dump($imageFilename);

// Newly created image file
var_dump($proxy->call($sessionId, 'product_media.list', 'Sku'));

$proxy->call($sessionId, 'product_media.update', array(
 'Sku',
 $imageFilename,
 array('position' => 2, 'types' => array('image')) /* Lets do it main image for
product */
));

// Updated image file
var_dump($proxy->call($sessionId, 'product_media.list', 'Sku'));

// Remove image file
$proxy->call($sessionId, 'product_media.remove', array('Sku', $imageFilename));

// Images without our file
var_dump($proxy->call($sessionId, 'product_media.list', 'Sku'));
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Create

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

### Media ^

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

## Module: Mage\_Catalog

Resource:catalog\_product\_attribute\_media

### Aliases:

- product\_attribute\_media
- product\_media

### Method:

- catalog\_product\_attribute\_media.create (SOAP V1)
- catalogProductAttributeMediaCreate (SOAP V2)

Allows you to upload a new product image.

### Aliases:

- product\_attribute\_media.create
- product\_media.create

### Arguments:

| Type   | Name           | Description                                                                |
|--------|----------------|----------------------------------------------------------------------------|
| string | sessionId      | Session ID                                                                 |
| string | product        | Product ID or code                                                         |
| array  | data           | Array of catalogProductAttributeMediaCreateEntity                          |
| string | storeView      | Store view ID or code (optional)                                           |
| string | identifierType | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

| Type   | Name   | Description                              |
|--------|--------|------------------------------------------|
| string | result | Image file name (e.g., "/i/m/image.png") |

The **catalogProductAttributeMediaCreateEntity** content is as follows:

| Type          | Name     | Description                                                               |
|---------------|----------|---------------------------------------------------------------------------|
| array         | file     | Array of catalogProductImageFileEntity                                    |
| string        | label    | Image label                                                               |
| string        | position | Image position                                                            |
| ArrayOfString | types    | Array of types                                                            |
| string        | exclude  | Defines whether the image will associate only to one of three image types |
| string        | remove   | Remove image flag                                                         |

The **catalogProductImageFileEntity** content is as follows:

| Type   | Name    | Description                        |
|--------|---------|------------------------------------|
| string | content | Image content (base_64 encoded)    |
| string | mime    | Image mime type (e.g., image/jpeg) |
| string | name    | Image name                         |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$productId = 2;
$file = array(
 'content' =>
 '/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIQIBAQICAgICAQAwUDAwMDAwYEBAMFBwYHBwcGBwcICQsJCAg
 'mime' => 'image/jpeg'
);

$result = $proxy->call(
 $session,
 'catalog_product_attribute_media.create',
 array(
 $productId,
 array('file'=>$file, 'label'=>'Label', 'position'=>'100',
 'types'=>array('thumbnail'), 'exclude'=>0)
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
```

```

$productId = 2;
$file = array(
 'content' =>
 '/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwcGBwcICQsJCAg
 'mime' => 'image/jpeg'
);

$result = $proxy->catalogProductAttributeMediaCreate(
 $sessionId,
 $productId,
 array('file' => $file, 'label' => 'Label', 'position' => '100', 'types' =>
array('thumbnail'), 'exclude' => 0)
);

```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductAttributeMediaCreate((object)array('sessionId' =>
$sessionId->result, 'productId' => '1', 'data' => ((object)array(
 'label' => 'image_label',
 'position' => '1',
 'types' => array('thumbnail'),
 'exclude' => '0',
 'file' => ((object)array(
 'content' =>
 '/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwcGBwcICQsJCAg
 'mime' => 'image/png',
 'name' => 'image'
)))));
 var_dump($result->result);

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Current Store

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

### Media ^

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

## Module: Mage\_Catalog

Resource:catalog\_product\_attribute\_media

### Aliases:

- product\_attribute\_media
- product\_media

### Method:

- catalog\_product\_attribute\_media.currentStore (SOAP V1)
- catalogProductAttributeMediaCurrentStore (SOAP V2)

Allows you to set/get the current store view.

### Aliases:

- product\_attribute\_media.currentStore
- product\_media.currentStore

### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | storeView | Store view ID or code (optional) |

### Returns:

| Type | Name      | Description   |
|------|-----------|---------------|
| int  | storeView | Store view ID |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_attribute_media.currentStore',
'english');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if
necessary

$result = $proxy->catalogProductAttributeMediaCurrentStore($sessionId, 'english');
var_dump($result);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Media Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

### Media ^

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

## Module: Mage\_Catalog

Resource:catalog\_product\_attribute\_media

### Aliases:

- product\_attribute\_media
- product\_media

### Method:

- catalog\_product\_attribute\_media.info (SOAP V1)
- catalogProductAttributeMediaInfo (SOAP V2)

Allows you to retrieve information about the specified product image.

### Aliases:

- product\_attribute\_media.info
- product\_media.info

### Arguments:

| Type   | Name              | Description                                                                |
|--------|-------------------|----------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                 |
| string | product\productId | Product ID or SKU                                                          |
| string | file              | Name of the image file (e.g., /b/l/blackberry8100_2.jpg)                   |
| string | storeView         | Store view ID or code (optional)                                           |
| string | identifierType    | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

| Type  | Name   | Description                        |
|-------|--------|------------------------------------|
| array | result | Array of catalogProductImageEntity |

The **catalogProductImageEntity** content is as follows:

| Type          | Name     | Description                                                               |
|---------------|----------|---------------------------------------------------------------------------|
| string        | file     | Image file name                                                           |
| string        | label    | Image file label                                                          |
| string        | position | Image file position                                                       |
| string        | exclude  | Defines whether the image will associate only to one of three image types |
| string        | url      | Image URL                                                                 |
| ArrayOfString | types    | Array of types                                                            |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_attribute_media.info',
array('product' => '2', 'file' => '/b/l/blackberry8100_2.jpg'));
var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeMediaInfo($sessionId, '2',
'/b/l/blackberry8100_2.jpg');
var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductAttributeMediaInfo((object)array('sessionId' =>
$sessionId->result, 'productId' => '1', 'file' => '/i/m/image.png'));

var_dump ($result->result);
```

## Response Example SOAP V1

```
array
 'file' => string '/b/l/blackberry8100_2.jpg' (length=25)
 'label' => string '' (length=0)
 'position' => string '1' (length=1)
 'exclude' => string '0' (length=1)
 'url' => string 'http://magentohost/media/catalog/product/b/l/blackberry8100_2.jpg'
 (length=71)
 'types' =>
 array
 0 => string 'image' (length=5)
 1 => string 'small_image' (length=11)
 2 => string 'thumbnail' (length=9)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Media List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

### Media ^

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

## Module: Mage\_Catalog

Resource:catalog\_product\_attribute\_media

### Aliases:

- product\_attribute\_media
- product\_media

### Method:

- catalog\_product\_attribute\_media.list (SOAP V1)
- catalogProductAttributeMediaList (SOAP V2)

Allows you to retrieve the list of product images.

### Aliases:

- product\_attribute\_media.list
- product\_media.list

### Arguments:

| Type   | Name              | Description                                                                |
|--------|-------------------|----------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                 |
| string | product\productId | Product ID or SKU                                                          |
| string | storeView         | Store view ID or code (optional)                                           |
| string | identifierType    | Defines whether the product ID or sku is passed in the 'product' parameter |

### Returns:

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

| Type  | Name   | Description                        |
|-------|--------|------------------------------------|
| array | result | Array of catalogProductImageEntity |

The **catalogProductImageEntity** content is as follows:

| Type          | Name     | Description                                                               |
|---------------|----------|---------------------------------------------------------------------------|
| string        | file     | Image file name                                                           |
| string        | label    | Image label                                                               |
| string        | position | Image position                                                            |
| string        | exclude  | Defines whether the image will associate only to one of three image types |
| string        | url      | Image URL                                                                 |
| ArrayOfString | types    | Array of types                                                            |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_attribute_media.list', '2');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeMediaList($sessionId, '2');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductAttributeMediaList((object)array('sessionId' => $sessionId->result, 'productId' => '2'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
'file' => string '/b/l/blackberry8100_2.jpg' (length=25)
'label' => string '' (length=0)
'position' => string '1' (length=1)
'exclude' => string '0' (length=1)
'url' => string 'http://magentopath/blackberry8100_2.jpg' (length=71)
'types' =>
array
0 => string 'image' (length=5)
1 => string 'small_image' (length=11)
2 => string 'thumbnail' (length=9)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Media Remove

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

### Media ^

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

## Module: Mage\_Catalog

Resource:catalog\_product\_attribute\_media

### Aliases:

- product\_attribute\_media
- product\_media

### Method:

- catalog\_product\_attribute\_media.remove (SOAP V1)
- catalogProductAttributeMediaRemove (SOAP V2)

Allows you to remove the image from a product.

### Aliases:

- product\_attribute\_media.remove
- product\_media.remove

### Arguments:

| Type   | Name              | Description                                                                |
|--------|-------------------|----------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                 |
| string | product\productId | Product ID or SKU                                                          |
| string | file              | Image file name (e.g., /b/l/blackberry8100_2.jpg)                          |
| string | identifierType    | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

| Type        | Description                                     |
|-------------|-------------------------------------------------|
| boolean int | True (1) if the image is removed from a product |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_attribute_media.remove',
array('product' => '3', 'file' => '/b/l/blackberry8100_2.jpg'));
var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeMediaRemove($sessionId, '3',
'/b/l/blackberry8100_2.jpg');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductAttributeMediaRemove((object)array('sessionId' =>
$sessionId->result, 'productId' => '3', 'file' => '/b/l/blackberry8100_2.jpg'));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Media Types

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▾

[Catalog Category Attributes](#) ▾

[Catalog Product](#) ▾

[Catalog Product Attribute](#) ▾

[Catalog Product Attribute](#)

## Media ^

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[Catalog Product Attribute Set](#) ▾

[Catalog Product Custom](#)

[Option](#) ▾

[Catalog Product Custom Option](#)

[Value](#) ▾

[Catalog Product Downloadable](#)

[Link](#) ▾

[Catalog Product Link](#) ▾

[Catalog Product Tag](#) ▾

[Catalog Product Tier Price](#) ▾

[Catalog Product Types](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

## Module: Mage\_Catalog

Resource:catalog\_product\_attribute\_media

### Aliases:

- product\_attribute\_media
- product\_media

### Method:

- catalog\_product\_attribute\_media.types (SOAP V1)
- catalogProductAttributeMediaTypes (SOAP V2)

Allows you to retrieve product image types including standard image, small\_image, thumbnail, etc. Note that if the product attribute set contains attributes of the Media Image type ([Catalog Input Type for Store Owner > Media Image](#)), it will also be returned in the response.

### Aliases:

- product\_attribute\_media.types
- product\_media.types

### Arguments:

| Type   | Name      | Description                     |
|--------|-----------|---------------------------------|
| string | sessionId | Session ID                      |
| string | setId     | ID of the product attribute set |

### Returns:

| Type | Name | Description |
|------|------|-------------|
|      |      |             |

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

|       |        |                                                 |
|-------|--------|-------------------------------------------------|
| array | result | Array of catalogProductAttributeMediaTypeEntity |
|-------|--------|-------------------------------------------------|

The **catalogProductAttributeMediaTypeEntity** content is as follows:

| Type   | Name  | Description                             |
|--------|-------|-----------------------------------------|
| string | code  | Image type code                         |
| string | scope | Image scope (store, website, or global) |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_attribute_media.types', '4');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeMediaTypes($sessionId, '4');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductAttributeMediaTypes((object)array('sessionId' => $sessionId->result, 'setId' => '4'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'code' => string 'image' (length=5)
 'scope' => string 'store' (length=5)
1 =>
array
 'code' => string 'small_image' (length=11)
 'scope' => string 'store' (length=5)
2 =>
```

```
array
 'code' => string 'thumbnail' (length=9)
 'scope' => string 'store' (length=5)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Media Update

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

### Media ^

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[catalog\\_product\\_attribute\\_me...](#)

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

## Module: Mage\_Catalog

Resource:catalog\_product\_attribute\_media

### Aliases:

- product\_attribute\_media
- product\_media

### Method:

- catalog\_product\_attribute\_media.update (SOAP V1)
- catalogProductAttributeMediaUpdate (SOAP V2)

Allows you to update the product image.

### Aliases:

- product\_attribute\_media.update
- product\_media.update

### Arguments:

| Type   | Name              | Description                                                                |
|--------|-------------------|----------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                 |
| string | product\productId | Product ID or code                                                         |
| string | file              | Image file name (e.g., /i/m/image.jpeg)                                    |
| array  | data              | Array of catalogProductAttributeMediaCreateEntity                          |
| string | storeView         | Store view ID or code                                                      |
| string | identifierType    | Defines whether the product ID or SKU is passed in the 'product' parameter |

|                                             |
|---------------------------------------------|
| <a href="#">Customer</a>                    |
| <a href="#">Directory</a>                   |
| <a href="#">Sales</a>                       |
| <a href="#">Enterprise Customer Balance</a> |
| <a href="#">Enterprise Gift Card</a>        |
| <a href="#">Enterprise Gift Message</a>     |
| <a href="#">Miscellaneous</a>               |
| <a href="#">WS-I Compliance</a>             |

**Notes:** You should specify only those parameters which you want to be updated. Parameters that were not specified in the request, will preserve the previous values.

#### Returns:

| Type    | Name   | Description                      |
|---------|--------|----------------------------------|
| boolean | result | Result of product image updating |

The **catalogProductAttributeMediaCreateEntity** content is as follows:

| Type          | Name     | Description                                                               |
|---------------|----------|---------------------------------------------------------------------------|
| array         | file     | Array of catalogProductImageFileEntity                                    |
| string        | label    | Product image label                                                       |
| string        | position | Product image position                                                    |
| ArrayOfString | types    | Array of types                                                            |
| string        | exclude  | Defines whether the image will associate only to one of three image types |
| string        | remove   | Image remove flag                                                         |

The **catalogProductImageFileEntity** content is as follows:

| Type   | Name    | Description                             |
|--------|---------|-----------------------------------------|
| string | content | Product image content (base_64 encoded) |
| string | mime    | Image mime type (e.g., image/jpeg)      |
| string | name    | Image name                              |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$productID = 1;
$file = '/i/m/image.jpg';

$newFile = array(
 'content' =>
 '/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwcGBwcICQsJCAg
 'mime' => 'image/jpeg'
);

$result = $client->call(
 $sessionId,
 'catalog_product_attribute_media.update',
 array(
 $productId,
 $file,
 array('file' => $newFile, 'label' => 'New label', 'position' => '50',
 'types' => array('image'), 'exclude' => 1)
)
);
```

```
)
);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$productId = 1;
$file = '/i/m/image.jpg';

$newFile = array(
 'content' =>
 '/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAIABAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwcGBwcICQsJCAg

 'mime' => 'image/jpeg'
);

$result = $client->catalogProductAttributeMediaUpdate(
 $sessionId,
 $productId,
 $file,
 array('file' => $newFile, 'label' => 'New label', 'position' => '50', 'types' =>
 array('image'), 'exclude' => 1)
);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
 'apiKey'));

$result = $proxy->catalogProductAttributeMediaUpdate((object)array('sessionId' =>
 $sessionId->result, 'productId' => '1', 'file' => '/t/u/tulips.jpg', 'data' =>
 ((object)array(
 'label' => 'tulips',
 'position' => '1',
 'remove' => '0',
 'types' => array('small_image')
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product Attribute Set

[Edit this page on GitHub](#) 

## Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

product\_attribute\_set.attribute...

product\_attribute\_set.attribute...

product\_attribute\_set.create

product\_attribute\_set.groupAdd

product\_attribute\_set.groupRe...

product\_attribute\_set.groupRe...

product\_attribute\_set.list

product\_attribute\_set.remove

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Product Attribute Sets

Allows you to retrieve product attribute sets.

**Resource Name:** catalog\_product\_attribute\_set

#### Aliases:

- product\_attribute\_set

#### Methods:

- [product\\_attribute\\_set.list](#) - Retrieve the list of product attribute sets
- [product\\_attribute\\_set.attributeAdd](#) - Add an attribute to the attribute set
- [product\\_attribute\\_set.attributeRemove](#) - Remove an attribute from an attribute set
- [product\\_attribute\\_set.create](#) - Create a new attribute set
- [product\\_attribute\\_set.groupAdd](#) - Add a new group for attributes in the attribute set
- [product\\_attribute\\_set.groupRemove](#) - Remove a group of attributes from an attribute set
- [product\\_attribute\\_set.groupRename](#) - Rename a group of attributes in an attribute set
- [product\\_attribute\\_set.remove](#) - Remove an attribute set

#### Faults:

| Fault Code | Fault Message                                                 |
|------------|---------------------------------------------------------------|
| 100        | Attribute set with requested id does not exist.               |
| 101        | Invalid data given.                                           |
| 102        | Error while creating attribute set. Details in error message. |

## Create Your Own API

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

|     |                                                                                                                                   |
|-----|-----------------------------------------------------------------------------------------------------------------------------------|
| 103 | Error while removing attribute set. Details in error message.                                                                     |
| 104 | Attribute set with requested id does not exist.                                                                                   |
| 105 | Unable to remove attribute set as it has related goods. Use forceProductsRemove parameter to remove attribute set with all goods. |
| 106 | Attribute with requested id does not exist.                                                                                       |
| 107 | Error while adding attribute to attribute set. Details in error message.                                                          |
| 108 | Attribute group with requested id does not exist.                                                                                 |
| 109 | Requested attribute is already in requested attribute set.                                                                        |
| 110 | Error while removing attribute from attribute set. Details in error message.                                                      |
| 111 | Requested attribute is not in requested attribute set.                                                                            |
| 112 | Requested group exist already in requested attribute set.                                                                         |
| 113 | Error while adding group to attribute set. Details in error message.                                                              |
| 114 | Error while renaming group. Details in error message.                                                                             |
| 115 | Error while removing group from attribute set. Details in error message.                                                          |
| 116 | Group can not be removed as it contains system attributes.                                                                        |
| 117 | Group can not be removed as it contains attributes, used in configurable products.                                                |

### Example:

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

echo "<pre>";
// create new set
setName = "New Test Set";
$skeletonId = 4;

setId = $proxy->call(
 $sessionId,
 "product_attribute_set.create",
 array(
 $setName,
 $skeletonId
)
);

// Get list
$list = $proxy->call(
 $sessionId,
 "product_attribute_set.list"
);
echo "Set list:\n";
print_r($list);

// create group
$groupName = "Test Group";
$groupId = $proxy->call(
 $sessionId,
```

```

"product_attribute_set.groupAdd",
array(
 $setId,
 $groupName
)
);

// rename group
$newGroupName = "New Test Group";
$result = $proxy->call(
 $sessionId,
 "product_attribute_set.groupRename",
array(
 $groupId,
 $newGroupName
)
);
;

// add attribute
$attributeId = 83;
$result = $proxy->call(
 $sessionId,
 "product_attribute_set.attributeAdd",
array(
 $attributeId,
 $setId
)
);
;

//remove attribute
$result = $proxy->call(
 $sessionId,
 "product_attribute_set.attributeRemove",
array(
 $attributeId,
 $setId
)
);
;

// remove group
$result = $proxy->call(
 $sessionId,
 "product_attribute_set.groupRemove",
array(
 $groupId
)
);
;

// remove set
$result = $proxy->call(
 $sessionId,
 "product_attribute_set.remove",
array(
 $setId
)
);
;
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute Set Add

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▲

[product\\_attribute\\_set.attributeAdd](#)

[product\\_attribute\\_set.attributeRemove](#)

[product\\_attribute\\_set.create](#)

[product\\_attribute\\_set.groupAdd](#)

[product\\_attribute\\_set.groupReplace](#)

[product\\_attribute\\_set.groupReplace](#)

[product\\_attribute\\_set.list](#)

[product\\_attribute\\_set.remove](#)

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

### Module: Product Attribute Set API

Resource: product\_attribute\_set

#### Method:

- `product_attribute_set.attributeAdd` (SOAP V1)
- `catalogProductAttributeSetAttributeAdd` (SOAP V2)

Allows you to add an existing attribute to an attribute set.

#### Arguments:

| Type   | Name             | Description           |
|--------|------------------|-----------------------|
| string | sessionId        | Session ID            |
| string | attributeId      | Attribute ID          |
| string | attributeSetId   | Attribute set ID      |
| string | attributeGroupId | Group ID (optional)   |
| string | sortOrder        | Sort order (optional) |

**Note:** If the `attributeGroupId` parameter is not passed, the attribute is added to the *General* group by default.

#### Returns:

| Type    | Name    | Description                                        |
|---------|---------|----------------------------------------------------|
| boolean | isAdded | True if the attribute is added to an attribute set |

#### Faults:

| Fault Code | Fault Message |
|------------|---------------|
|------------|---------------|

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

|     |                                                                          |
|-----|--------------------------------------------------------------------------|
| 104 | Attribute set with requested id does not exist.                          |
| 106 | Attribute with requested id does not exist.                              |
| 107 | Error while adding attribute to attribute set. Details in error message. |
| 108 | Attribute group with requested id does not exist.                        |
| 109 | Requested attribute is already in requested attribute set.               |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$setId = 5;
$attributeId = 83;

$result = $proxy->call(
 $sessionId,
 "product_attribute_set.attributeAdd",
 array(
 $attributeId,
 $setId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$setId = 5;
$attributeId = 83;

$result = $proxy->catalogProductAttributeSetAttributeAdd(
 $sessionId,
 $attributeId,
 $setId
);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductAttributeSetAttributeAdd((object)array('sessionId' => $sessionId->result, 'attributeId' => '5', 'attributeSetId' => '83'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Attribute Remove

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▲

[product\\_attribute\\_set.attribute...](#)

[product\\_attribute\\_set.attribute...](#)

[product\\_attribute\\_set.create](#)

[product\\_attribute\\_set.groupAdd](#)

[product\\_attribute\\_set.groupRe...](#)

[product\\_attribute\\_set.groupRe...](#)

[product\\_attribute\\_set.list](#)

[product\\_attribute\\_set.remove](#)

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

## Module: Product Attribute Set API

Resource: `product_attribute_set`

### Method:

- `product_attribute_set.attributeRemove` (SOAP V1)
- `catalogProductAttributeSetAttributeRemove` (SOAP V2)

Allows you to remove an existing attribute from an attribute set.

### Arguments:

| Type   | Name           | Description      |
|--------|----------------|------------------|
| string | sessionId      | Session ID       |
| string | attributeId    | Attribute ID     |
| string | attributeSetId | Attribute set ID |

### Returns:

| Type    | Name      | Description                                            |
|---------|-----------|--------------------------------------------------------|
| boolean | isRemoved | True if the attribute is removed from an attribute set |

### Faults:

| Fault Code | Fault Message                                                                |
|------------|------------------------------------------------------------------------------|
| 104        | Attribute set with requested id does not exist.                              |
| 106        | Attribute with requested id does not exist.                                  |
| 110        | Error while removing attribute from attribute set. Details in error message. |

## Create Your Own API

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

111

Requested attribute is not in requested attribute set.

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$setId = 5;
$attributeId = 83;

$result = $proxy->call(
 $sessionId,
 "product_attribute_set.attributeRemove",
 array(
 $attributeId,
 $setId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeSetAttributeRemove($sessionId, '5', '83');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductAttributeSetAttributeRemove((object)array('sessionId' => $sessionId->result, 'attributeId' => '5', 'attributeSetId' => '83'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute Set Create

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▲

product\_attribute\_set.attribute...

product\_attribute\_set.attribute...

[product\\_attribute\\_set.create](#)

product\_attribute\_set.groupAdd

product\_attribute\_set.groupRe...

product\_attribute\_set.groupRe...

product\_attribute\_set.list

product\_attribute\_set.remove

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

### Module: Product Attribute Set API

Resource: product\_attribute\_set

#### Method:

- `product_attribute_set.create` (SOAP V1)
- `catalogProductAttributeSetCreate` (SOAP V2)

Allows you to create a new attribute set based on another attribute set.

#### Arguments:

| Type   | Name             | Description                                                            |
|--------|------------------|------------------------------------------------------------------------|
| string | sessionId        | Session ID                                                             |
| string | attributeSetName | Attribute set name                                                     |
| string | skeletonSetId    | Attribute set ID basing on which the new attribute set will be created |

#### Return:

| Type | Name  | Description                     |
|------|-------|---------------------------------|
| int  | setId | ID of the created attribute set |

#### Faults:

| Fault Code | Fault Message                                                 |
|------------|---------------------------------------------------------------|
| 100        | Attribute set with requested id does not exist.               |
| 101        | Invalid data given.                                           |
| 102        | Error while creating attribute set. Details in error message. |

[Customer](#) ▾[Directory](#) ▾[Sales](#) ▾[Enterprise Customer Balance](#) ▾[Enterprise Gift Card](#) ▾[Enterprise Gift Message](#) ▾[Miscellaneous](#) ▾[WS-I Compliance](#)

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$setName = "New Attribute Set";
$sessionId = 4;

$newsetId = $proxy->call(
 $sessionId,
 "product_attribute_set.create",
 array(
 $setName,
 $sessionId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$attributeSetName = 'New Attribute Set';
$sessionId = 4;

$result = $client->catalogProductAttributeSetCreate(
 $sessionId,
 $attributeSetName,
 $sessionId
);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductAttributeSetCreate((object)array('sessionId' =>
$sessionId->result, 'attributeSetName' => 'New Attribute Set', 'skeletonSetId' =>
'4'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute Set Group Add

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▲

[product\\_attribute\\_set.attribute...](#)

[product\\_attribute\\_set.attribute...](#)

[product\\_attribute\\_set.create](#)

[product\\_attribute\\_set.groupAdd](#)

[product\\_attribute\\_set.groupRe...](#)

[product\\_attribute\\_set.groupRe...](#)

[product\\_attribute\\_set.list](#)

[product\\_attribute\\_set.remove](#)

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

### Module: Product Attribute Set API

Resource: product\_attribute\_set

#### Method:

- `product_attribute_set.groupAdd` (SOAP V1)
- `catalogProductAttributeSetGroupAdd` (SOAP V2)

Allows you to add a new group for attributes to the attribute set.

#### Arguments:

| Type   | Name           | Description      |
|--------|----------------|------------------|
| string | sessionId      | Session ID       |
| string | attributeSetId | Attribute set ID |
| string | groupName      | Group name       |

#### Return:

| Type | Name   | Description             |
|------|--------|-------------------------|
| int  | result | ID of the created group |

#### Faults:

| Fault Code | Fault Message                                                        |
|------------|----------------------------------------------------------------------|
| 112        | Requested group exist already in requested attribute set.            |
| 113        | Error while adding group to attribute set. Details in error message. |

## Create Your Own API

Customer ▾

Directory ▾

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'product_attribute_set.groupAdd',
array('attributeSetId' => '9', 'groupName' => 'new_group'));
var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeSetGroupAdd($sessionId, '9', 'new_group');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://maentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductAttributeSetGroupAdd((object)array('sessionId' =>
$sessionId->result, 'attributeSetId' => '9', 'groupName' => 'new_group'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute Set Group Remove

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▲

product\_attribute\_set.attribute...

product\_attribute\_set.attribute...

product\_attribute\_set.create

product\_attribute\_set.groupAdd

[product\\_attribute\\_set.groupRe...](#)

product\_attribute\_set.groupRe...

product\_attribute\_set.list

product\_attribute\_set.remove

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

### Module: Product Attribute Set API

Resource: product\_attribute\_set

#### Method:

- product\_attribute\_set.groupRemove (SOAP V1)
- catalogProductAttributeSetGroupRemove (SOAP V2)

Allows you to remove a group from an attribute set.

#### Arguments:

| Type   | Name             | Description |
|--------|------------------|-------------|
| string | sessionId        | Session ID  |
| string | attributeGroupId | Group ID    |

#### Return:

| Type        | Description                      |
|-------------|----------------------------------|
| boolean/int | True (1) if the group is removed |

#### Faults:

| Fault Code | Fault Message                                                                      |
|------------|------------------------------------------------------------------------------------|
| 108        | Attribute group with requested id does not exist.                                  |
| 115        | Error while removing group from attribute set. Details in error message.           |
| 116        | Group can not be removed as it contains system attributes.                         |
| 117        | Group can not be removed as it contains attributes, used in configurable products. |

[Customer](#) ▾[Directory](#) ▾[Sales](#) ▾[Enterprise Customer Balance](#) ▾[Enterprise Gift Card](#) ▾[Enterprise Gift Message](#) ▾[Miscellaneous](#) ▾[WS-I Compliance](#)

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$groupId = 70;

$result = $proxy->call(
 $sessionId,
 "product_attribute_set.groupRemove",
 array(
 $groupId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeSetGroupRemove($sessionId, '70');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductAttributeSetGroupRemove((object)array('sessionId' => $sessionId->result, 'attributeGroupId' => '70'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Attribute Set Group Rename

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▲

[product\\_attribute\\_set.attribute...](#)

[product\\_attribute\\_set.attribute...](#)

[product\\_attribute\\_set.create](#)

[product\\_attribute\\_set.groupAdd](#)

[product\\_attribute\\_set.groupRe...](#)

[product\\_attribute\\_set.groupRe...](#)

[product\\_attribute\\_set.list](#)

[product\\_attribute\\_set.remove](#)

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

## Module: Product Attribute Set API

Resource: product\_attribute\_set

### Method:

- [product\\_attribute\\_set.groupRename \(SOAP V1\)](#)
- [catalogProductAttributeSetGroupRename \(SOAP V2\)](#)

Allows you to rename a group in the attribute set.

### Arguments:

| Type   | Name      | Description                          |
|--------|-----------|--------------------------------------|
| string | sessionId | Session ID                           |
| string | groupId   | ID of the group that will be renamed |
| string | groupName | New name for the group               |

### Return:

| Type        | Description                      |
|-------------|----------------------------------|
| boolean int | True (1) if the group is renamed |

### Faults:

| Fault Code | Fault Message                                         |
|------------|-------------------------------------------------------|
| 108        | Attribute group with requested id does not exist.     |
| 114        | Error while renaming group. Details in error message. |

[Customer](#) ▾[Directory](#) ▾[Sales](#) ▾[Enterprise Customer Balance](#) ▾[Enterprise Gift Card](#) ▾[Enterprise Gift Message](#) ▾[Miscellaneous](#) ▾[WS-I Compliance](#)

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$groupId = 100;
$groupName = "New Group";

echo "Renaming group...";
$result = $proxy->call(
 $sessionId,
 "product_attribute_set.groupRename",
 array(
 $groupId,
 $groupName
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeSetGroupRename($sessionId, '100', 'New Group');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductAttributeSetGroupRename((object)array('sessionId' => $sessionId->result, 'groupId' => '100', 'groupName' => 'New Group'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute Set List

[Edit this page on GitHub](#) 

### Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

product\_attribute\_set.attribute...

product\_attribute\_set.attribute...

product\_attribute\_set.create

product\_attribute\_set.groupAdd

product\_attribute\_set.groupRe...

product\_attribute\_set.groupRe...

[product\\_attribute\\_set.list](#)

product\_attribute\_set.remove

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

## Module: Mage\_Catalog

Resource: catalog\_product\_attribute\_set

### Aliases:

- product\_attribute\_set

### Method:

- catalog\_product\_attribute\_set.list (SOAP V1)
- catalogProductAttributeSetList (SOAP V2)

Allows you to retrieve the list of product attribute sets.

### Aliases:

- product\_attribute\_set.list

### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

### Returns:

| Type  | Name   | Description                               |
|-------|--------|-------------------------------------------|
| array | result | Array of catalogProductAttributeSetEntity |

The **catalogProductAttributeSetEntity** content is as follows:

| Type | Name   | Description             |
|------|--------|-------------------------|
| int  | set_id | ID of the attribute set |

## Create Your Own API

Customer ▾

Directory ▾

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

|        |      |                    |
|--------|------|--------------------|
| string | name | Attribute set name |
|--------|------|--------------------|

### Faults:

No faults.

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_attribute_set.list');
var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeSetList($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductAttributeSetList((object)array('sessionId' =>
$sessionId->result));
var_dump($result->result);
```

## Response Example SOAP V1

```
array
0 =>
array
 'set_id' => string '4' (length=1)
 'name' => string 'Default' (length=7)
1 =>
array
 'set_id' => string '9' (length=1)
 'name' => string 'products_set' (length=12)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Attribute Set Remove

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▲

[product\\_attribute\\_set.attribute...](#)

[product\\_attribute\\_set.attribute...](#)

[product\\_attribute\\_set.create](#)

[product\\_attribute\\_set.groupAdd](#)

[product\\_attribute\\_set.groupRe...](#)

[product\\_attribute\\_set.groupRe...](#)

[product\\_attribute\\_set.list](#)

[product\\_attribute\\_set.remove](#)

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

### Module: Product Attribute Set API

#### Resource: product\_attribute\_set

##### Method:

- `product_attribute_set.remove` (SOAP V1)
- `catalogProductAttributeSetRemove` (SOAP V2)

Allows you to remove an existing attribute set.

##### Arguments:

| Type   | Name                | Description                          |
|--------|---------------------|--------------------------------------|
| string | sessionId           | Session ID                           |
| string | attributeSetId      | Attribute set ID                     |
| string | forceProductsRemove | Force product remove flag (optional) |

##### Return:

| Type        | Name      | Description                              |
|-------------|-----------|------------------------------------------|
| boolean/int | isRemoved | True (1) if the attribute set is removed |

##### Faults:

| Fault Code | Fault Message                                                 |
|------------|---------------------------------------------------------------|
| 103        | Error while removing attribute set. Details in error message. |
| 104        | Attribute set with requested id does not exist.               |

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

105

Unable to remove attribute set as it has related goods. Use forceProductsRemove parameter to remove attribute set with all goods.

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$setId = 5;

$result = $proxy->call(
 $sessionId,
 "product_attribute_set.remove",
 array(
 $setId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductAttributeSetRemove($sessionId, '5');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductAttributeSetRemove((object)array('sessionId' => $sessionId->result, 'attributeSetId' => '5'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Catalog Product Custom Option

[Edit this page on GitHub](#) 

### Introduction

#### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

#### Option ^

[product\\_custom\\_option.add](#)

[product\\_custom\\_option.info](#)

[product\\_custom\\_option.list](#)

[product\\_custom\\_option.remove](#)

[product\\_custom\\_option.types](#)

[product\\_custom\\_option.update](#)

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

## Module: Mage\_Catalog

Resource: catalog\_product\_custom\_option

**Aliases:** product\_custom\_option

### Methods:

- [product\\_custom\\_option.add](#) - Add a new custom option to a product
- [product\\_custom\\_option.update](#) - Update the product custom option
- [product\\_custom\\_option.types](#) - Get the list of available custom option types
- [product\\_custom\\_option.list](#) - Retrieve the list of product custom options
- [product\\_custom\\_option.info](#) - Get full information about the custom option in a product
- [product\\_custom\\_option.remove](#) - Remove the custom option

### Faults:

| Fault Code | Fault Message                                                                   |
|------------|---------------------------------------------------------------------------------|
| 101        | Product with requested id does not exist.                                       |
| 102        | Provided data is invalid.                                                       |
| 103        | Error while saving an option. Details are in the error message.                 |
| 104        | Store with requested code/id does not exist.                                    |
| 105        | Option with requested id does not exist.                                        |
| 106        | Invalid option type provided. Call 'types' to get list of allowed option types. |
| 107        | Error while deleting an option. Details are in the error message.               |

Create the Magento file system owner

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

---

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Option Add

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ^

[product\\_custom\\_option.add](#)

[product\\_custom\\_option.info](#)

[product\\_custom\\_option.list](#)

[product\\_custom\\_option.remove](#)

[product\\_custom\\_option.types](#)

[product\\_custom\\_option.update](#)

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

Module: Complex Product API

Resource: product\_custom\_option

### Method:

- `product_custom_option.add` (SOAP V1)
- `catalogProductCustomOptionAdd` (SOAP V2)

Allows you to add a new custom option for a product.

### Arguments:

| Type   | Name      | Description                              |
|--------|-----------|------------------------------------------|
| string | sessionId | Session ID                               |
| string | productId | Product ID                               |
| array  | data      | Array of catalogProductCustomOptionToAdd |
| string | store     | Store view ID or code (optional)         |

### Return:

| Type    | Description                        |
|---------|------------------------------------|
| boolean | True if the custom option is added |

The `catalogProductCustomOptionToAdd` content is as follows:

| Type   | Name  | Description  |
|--------|-------|--------------|
| string | title | Option title |
| string | type  | Option type  |

**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

|        |                   |                                                     |
|--------|-------------------|-----------------------------------------------------|
| string | sort_order        | Option sort order                                   |
| int    | is_require        | Defines whether the option is required              |
| array  | additional_fields | Array of catalogProductCustomOptionAdditionalFields |

The **catalogProductCustomOptionAdditionalFieldsEntity** content is as follows:

| Type   | Name           | Description                                                                      |
|--------|----------------|----------------------------------------------------------------------------------|
| string | title          | Custom option title                                                              |
| string | price          | Custom option price                                                              |
| string | price_type     | Price type. Possible values are as follows: "fixed" or "percent"                 |
| string | sku            | Custom option SKU                                                                |
| string | max_characters | Maximum number of characters for the customer input on the frontend (optional)   |
| string | sort_order     | Custom option sort order                                                         |
| string | file_extension | List of file extensions allowed to upload by the user on the frontend (optional) |
| string | image_size_x   | Width limit for uploaded images (optional)                                       |
| string | image_size_y   | Height limit for uploaded images (optional)                                      |
| string | value_id       | Value ID                                                                         |

**Faults:**

| Fault Code | Fault Message                                                                   |
|------------|---------------------------------------------------------------------------------|
| 101        | Product with requested id does not exist.                                       |
| 102        | Provided data is invalid.                                                       |
| 103        | Error while saving an option. Details are in the error message.                 |
| 104        | Store with requested code/id does not exist.                                    |
| 106        | Invalid option type provided. Call 'types' to get list of allowed option types. |

## Examples

**Request Example SOAP V1**

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$productID = 1;// Existing product ID

// Add custom option of Text Field type
$customTextFieldOption = array(
 "title" => "Custom Text Field Option Title",
 "type" => "field",
 "is_require" => 1,
 "sort_order" => 0,
 "additional_fields" => array(
 array(
 "price" => 10.00,
```

```

 "price_type" => "fixed",
 "sku" => "custom_text_option_sku",
 "max_characters" => 255
)
)
);
$resultCustomTextFieldOptionAdd = $proxy->call(
 $sessionId,
 "product_custom_option.add",
 array(
 $productId,
 $customTextFieldOption
)
);
// Add custom option of File type
$customFileOption = array(
 "title" => "Custom File Option Title",
 "type" => "file",
 "is_require" => 1,
 "sort_order" => 5,
 "additional_fields" => array(
 array(
 "price" => 10.00,
 "price_type" => "fixed",
 "sku" => "custom_file_option_sku",
 "file_extension" => "jpg",
 "image_size_x" => 800,
 "image_size_y" => 600
)
)
);
$resultCustomFileOptionAdd = $proxy->call(
 $sessionId,
 "product_custom_option.add",
 array(
 $productId,
 $customFileOption
)
);
// Add custom option of Dropdown type
$customDropdownOption = array(
 "title" => "Custom Dropdown Option Title",
 "type" => "drop_down",
 "is_require" => 1,
 "sort_order" => 10,
 "additional_fields" => array(
 array(
 "title" => "Dropdown row #1",
 "price" => 10.00,
 "price_type" => "fixed",
 "sku" => "custom_select_option_sku_1",
 "sort_order" => 0
),
 array(
 "title" => "Dropdown row #2",
 "price" => 10.00,
 "price_type" => "fixed",
 "sku" => "custom_select_option_sku_2",
 "sort_order" => 5
)
)
);

```

```

);
$resultCustomDropdownOptionAdd = $proxy->call(
 $sessionId,
 "product_custom_option.add",
 array(
 $productId,
 $customDropdownOption
)
);

```

## Request Example SOAP V2

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->catalogProductCustomOptionAdd($sessionId, '1', array('title' =>
'title',
'type' => 'field',
'sort_order' => '1',
'is_require' => 1,
'additional_fields' => array(array(
'price' => '15',
'price_type' => 'fixed',
'sku' => 'sku',
'max_characters' => '100'
)));
var_dump($result);

```

## Request Example SOAP V2 (WS-I Compliance Example)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductCustomOptionAdd((object)array('sessionId' =>
$sessionId->result, 'productId' => '1', 'store' => '1', 'data' => ((object)array(
'title' => 'title',
'type' => 'field',
'sort_order' => '1',
'is_require' => 1,
'additional_fields' => array(array(
'price' => '15',
'price_type' => 'fixed',
'sku' => 'sku',
'max_characters' => '100'
)))));
var_dump($result->result);

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Option Info

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ^

[product\\_custom\\_option.add](#)

[product\\_custom\\_option.info](#)

[product\\_custom\\_option.list](#)

[product\\_custom\\_option.remove](#)

[product\\_custom\\_option.types](#)

[product\\_custom\\_option.update](#)

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

Module: Complex Product API

Resource: product\_custom\_option

### Method:

- `product_custom_option.info` (SOAP V1)
- `catalogProductCustomOptionInfo` (SOAP V2)

Allows you to retrieve full information about the custom option in a product.

### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | optionId  | Option ID                        |
| string | store     | Store view ID or code (optional) |

### Return:

| Type  | Name   | Description                                   |
|-------|--------|-----------------------------------------------|
| array | result | Array of catalogProductCustomOptionInfoEntity |

The `catalogProductCustomOptionInfoEntity` content is as follows:

| Type   | Name       | Description                                                                    |
|--------|------------|--------------------------------------------------------------------------------|
| string | title      | Custom option title                                                            |
| string | type       | Custom option type. Can have one of the following values: "fixed" or "percent" |
| string | sort_order | Custom option sort order                                                       |

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

|       |                   |                                                     |
|-------|-------------------|-----------------------------------------------------|
| int   | is_require        | Defines whether the custom option is required       |
| array | additional_fields | Array of catalogProductCustomOptionAdditionalFields |

The **catalogProductCustomOptionAdditionalFields** content is as follows:

| Type   | Name           | Description                                                                                                      |
|--------|----------------|------------------------------------------------------------------------------------------------------------------|
| string | title          | Custom option title                                                                                              |
| string | price          | Custom option price                                                                                              |
| string | price_type     | Price type. Possible values are as follows: "fixed" or "percent"                                                 |
| string | sku            | Custom option SKU                                                                                                |
| string | max_characters | Maximum number of characters for the customer input on the frontend (optional)                                   |
| string | sort_order     | Custom option sort order                                                                                         |
| string | file_extension | List of file extensions allowed to upload by the user on the frontend (optional; for the <b>File</b> input type) |
| string | image_size_x   | Width limit for uploaded images (optional; for the <b>File</b> input type)                                       |
| string | image_size_y   | Height limit for uploaded images (optional; for the <b>File</b> input type)                                      |
| string | value_id       | Value ID                                                                                                         |

#### Faults:

| Fault Code | Fault Message                                |
|------------|----------------------------------------------|
| 101        | Product with requested id does not exist.    |
| 104        | Store with requested code/id does not exist. |
| 105        | Option with requested id does not exist.     |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'product_custom_option.info', '1');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary
```

```
$result = $proxy->catalogProductCustomOptionInfo($sessionId, '1');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductCustomOptionInfo((object)array('sessionId' =>
$sessionId->result, 'optionId' => '1'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 'title' => string 'model' (length=5)
 'type' => string 'drop_down' (length=9)
 'is_require' => string '1' (length=1)
 'sort_order' => string '0' (length=1)
 'additional_fields' =>
 array
 0 =>
 array
 'value_id' => string '1' (length=1)
 'title' => string 'monoblock' (length=9)
 'price' => string '139.9900' (length=8)
 'price_type' => string 'fixed' (length=5)
 'sku' => string 'monoblock' (length=9)
 'sort_order' => string '0' (length=1)
 1 =>
 array
 'value_id' => string '2' (length=1)
 'title' => string 'slider' (length=6)
 'price' => string '239.9900' (length=8)
 'price_type' => string 'fixed' (length=5)
 'sku' => string 'slider' (length=6)
 'sort_order' => string '0' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Option List

[Edit this page on GitHub](#) 

## Introduction

## Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[product\\_custom\\_option.add](#)

[product\\_custom\\_option.info](#)

[product\\_custom\\_option.list](#)

[product\\_custom\\_option.remove](#)

[product\\_custom\\_option.types](#)

[product\\_custom\\_option.update](#)

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

Module: Complex Product API

Resource: product\_custom\_option

### Method:

- `product_custom_option.list` (SOAP V1)
- `catalogProductCustomOptionList` (SOAP V2)

Allows you to retrieve the list of custom options for a specific product.

### Arguments:

| Type   | Name      | Description                                                 |
|--------|-----------|-------------------------------------------------------------|
| string | sessionId | Session ID                                                  |
| string | productId | Product ID                                                  |
| string | store     | Store view ID or code (optional but required for WS-I mode) |

### Return:

| Type  | Name   | Description                             |
|-------|--------|-----------------------------------------|
| array | result | Array of catalogProductCustomOptionList |

The `catalogProductCustomOptionList` content is as follows:

| Type   | Name      | Description         |
|--------|-----------|---------------------|
| string | option_id | Custom option ID    |
| string | title     | Custom option title |
| string | type      | Custom option type  |

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

|        |            |                                               |
|--------|------------|-----------------------------------------------|
| string | sort_order | Custom option sort order                      |
| int    | is_require | Defines whether the custom option is required |

#### Faults:

| Fault Code | Fault Message                                |
|------------|----------------------------------------------|
| 101        | Product with requested id does not exist.    |
| 104        | Store with requested code/id does not exist. |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'product_custom_option.list', '1');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductCustomOptionList($sessionId, '1');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductCustomOptionList((object)array('sessionId' => $sessionId->result, 'productId' => '1', 'store' => '1'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'option_id' => string '1' (length=1)
 'title' => string 'model' (length=5)
 'type' => string 'drop_down' (length=9)
 'is_require' => string '1' (length=1)
 'sort_order' => string '0' (length=1)
```



## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Option Remove

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ^

[product\\_custom\\_option.add](#)

[product\\_custom\\_option.info](#)

[product\\_custom\\_option.list](#)

[product\\_custom\\_option.remove](#)

[product\\_custom\\_option.types](#)

[product\\_custom\\_option.update](#)

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

Module: Complex Product API

Resource: product\_custom\_option

### Method:

- `product_custom_option.remove` (SOAP V1)
- `catalogProductCustomOptionRemove` (SOAP V2)

Allows you to remove a custom option from the product.

### Arguments:

| Type   | Name      | Description      |
|--------|-----------|------------------|
| string | sessionId | Session ID       |
| string | optionId  | Custom option ID |

### Return:

| Type    | Description                          |
|---------|--------------------------------------|
| boolean | True if the custom option is removed |

### Faults:

| Fault Code | Fault Message                                                     |
|------------|-------------------------------------------------------------------|
| 105        | Option with requested id does not exist.                          |
| 107        | Error while deleting an option. Details are in the error message. |

## Examples

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

---

## Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$optionId = 1;// Existing option ID

$result = $proxy->call(
 $sessionId,
 "product_custom_option.remove",
 array(
 $optionId
)
);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductCustomOptionRemove($sessionId, '1');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductCustomOptionRemove((object)array('sessionId' => $sessionId->result, 'optionId' => '1'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Option Types

[Edit this page on GitHub](#) 

## Introduction

## Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

## Option

[product\\_custom\\_option.add](#)

[product\\_custom\\_option.info](#)

[product\\_custom\\_option.list](#)

[product\\_custom\\_option.remove](#)

[product\\_custom\\_option.types](#)

[product\\_custom\\_option.update](#)

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

Module: Complex Product API

Resource: product\_custom\_option

### Method:

- [product\\_custom\\_option.types](#) (SOAP V1)
- [catalogProductCustomOptionTypes](#) (SOAP V2)

Allows you to retrieve the list of available custom option types.

### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

### Return:

| Type  | Name   | Description                              |
|-------|--------|------------------------------------------|
| array | result | Array of catalogProductCustomOptionTypes |

The **catalogProductCustomOptionTypesEntity** content is as follows:

| Type   | Name  | Description         |
|--------|-------|---------------------|
| string | label | Custom option label |
| string | value | Custom option value |

### Faults:

*No faults*

## Directory ▾

### Sales ▾

#### Enterprise Customer Balance ▾

#### Enterprise Gift Card ▾

#### Enterprise Gift Message ▾

#### Miscellaneous ▾

#### WS-I Compliance

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'product_custom_option.types');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductCustomOptionTypes($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductCustomOptionTypes((object)array('sessionId' => $sessionId->result));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'label' => string 'Field' (length=5)
 'value' => string 'field' (length=5)
1 =>
array
 'label' => string 'Area' (length=4)
 'value' => string 'area' (length=4)
2 =>
array
 'label' => string 'File' (length=4)
 'value' => string 'file' (length=4)
3 =>
array
 'label' => string 'Drop-down' (length=9)
 'value' => string 'drop_down' (length=9)
4 =>
array
 'label' => string 'Radio Buttons' (length=13)
 'value' => string 'radio' (length=5)
5 =>
```

```
array
 'label' => string 'Checkbox' (length=8)
 'value' => string 'checkbox' (length=8)
6 =>
 array
 'label' => string 'Multiple Select' (length=15)
 'value' => string 'multiple' (length=8)
7 =>
 array
 'label' => string 'Date' (length=4)
 'value' => string 'date' (length=4)
8 =>
 array
 'label' => string 'Date & Time' (length=11)
 'value' => string 'date_time' (length=9)
9 =>
 array
 'label' => string 'Time' (length=4)
 'value' => string 'time' (length=4)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Option Update

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

## Option ^

[product\\_custom\\_option.add](#)

[product\\_custom\\_option.info](#)

[product\\_custom\\_option.list](#)

[product\\_custom\\_option.remove](#)

[product\\_custom\\_option.types](#)

[product\\_custom\\_option.update](#)

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

## Module: Complex Product API

Resource: `product_custom_option`

### Method:

- `product_custom_option.update` (SOAP V1)
- `catalogProductCustomOptionUpdate` (SOAP V2)

Allows you to update the required product custom option.

### Arguments:

| Type   | Name      | Description                                 |
|--------|-----------|---------------------------------------------|
| string | sessionId | Session ID                                  |
| string | optionId  | Option ID                                   |
| array  | data      | Array of catalogProductCustomOptionToUpdate |
| string | store     | Store view ID or code (optional)            |

### Return:

| Type        | Description                              |
|-------------|------------------------------------------|
| boolean\int | True (1) if the custom option is updated |

The `catalogProductCustomOptionToUpdate` content is as follows:

| Type   | Name  | Description                              |
|--------|-------|------------------------------------------|
| string | title | Title of the custom option to be updated |
| string | type  | Custom option type                       |

**Directory** ▾**Sales** ▾**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

|        |                   |                                                     |
|--------|-------------------|-----------------------------------------------------|
| string | sort_order        | Custom option sort order                            |
| int    | is_require        | Defines whether the custom option is required       |
| array  | additional_fields | Array of catalogProductCustomOptionAdditionalFields |

The **catalogProductCustomOptionAdditionalFields** content is as follows:

| Type   | Name           | Description                                                                                                      |
|--------|----------------|------------------------------------------------------------------------------------------------------------------|
| string | title          | Custom option title                                                                                              |
| string | price          | Custom option price                                                                                              |
| string | price_type     | Price type. Possible values are as follows: "fixed" or "percent"                                                 |
| string | sku            | Custom option SKU                                                                                                |
| string | max_characters | Maximum number of characters for the customer input on the frontend (optional)                                   |
| string | sort_order     | Custom option sort order                                                                                         |
| string | file_extension | List of file extensions allowed to upload by the user on the frontend (optional; for the <b>File</b> input type) |
| string | image_size_x   | Width limit for uploaded images (optional; for the <b>File</b> input type)                                       |
| string | image_size_y   | Height limit for uploaded images (optional; for the <b>File</b> input type)                                      |
| string | value_id       | Value ID                                                                                                         |

**Faults:**

| Fault Code | Fault Message                                                                   |
|------------|---------------------------------------------------------------------------------|
| 101        | Product with requested id does not exist.                                       |
| 102        | Provided data is invalid.                                                       |
| 103        | Error while saving an option. Details are in the error message.                 |
| 104        | Store with requested code/id does not exist.                                    |
| 105        | Option with requested id does not exist.                                        |
| 106        | Invalid option type provided. Call 'types' to get list of allowed option types. |

**Examples****Request Example SOAP V1**

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$selectOptionId = 1379;
$selectOptionValueId = 794;
$textOptionId = 1380;
$fileOptionId = 1381;

// Update custom option of Text Field type
$customTextFieldOption = array(
 "title" => "Custom Text Field Option Title Updated",
 "value_id" => 1381,
 "sort_order" => 1,
 "price" => 100,
 "price_type" => "fixed",
 "sku" => "CustomTextFieldOption"
);
```

```

 "type" => "field",
 "is_require" => 1,
 "sort_order" => 20,
 "additional_fields" => array(
 array(
 "price" => 13.00,
 "price_type" => "fixed",
 "sku" => "custom_text_option_sku_updated",
 "max_characters" => 127
)
)
);
 $resultCustomTextFieldOptionUpdate = $proxy->call(
 $sessionId,
 "product_custom_option.update",
 array(
 $textOptionId,
 $customTextFieldOption
)
);
}

// Update custom option of File type
$customFileOption = array(
 "title" => "Custom File Option Title Updated",
 "additional_fields" => array(
 array(
 "image_size_x" => 800,
 "image_size_y" => 999
)
)
);
$resultCustomFileOptionUpdate = $proxy->call(
 $sessionId,
 "product_custom_option.update",
 array(
 $fileOptionId,
 $customFileOption
)
);

// Update custom option of Dropdown type
$customDropdownOption = array(
 "title" => "Custom Dropdown Option Title Updated to Multiselect",
 "type" => "multiple",
 "additional_fields" => array(
 array(
 "value_id" => $selectOptionValueId,
 "price" => 14.00,
 "price_type" => 'percent',
 "sku" => "custom_select_option_sku_1 updated",
 "sort_order" => 26
)
)
);
$resultCustomDropdownOptionUpdate = $proxy->call(
 $sessionId,
 "product_custom_option.update",
 array(
 $selectOptionId,
 $customDropdownOption
)
);

```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->catalogProductCustomOptionUpdate($sessionId, '1', array(
 'title' => 'title_updated',
 'is_require' => 0,
 'sort_order' => '2'
));
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductCustomOptionUpdate((object)array('sessionId' =>
$sessionId->result, 'optionId' => '1', 'data' => ((object)array(
 'title' => 'title_updated',
 'is_require' => 0,
 'sort_order' => '2'
))));
```

var\_dump(\$result->result);

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product Custom Option Value

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ^

product\_custom\_option\_value....  
product\_custom\_option\_value.i...  
product\_custom\_option\_value.list  
product\_custom\_option\_value....  
product\_custom\_option\_value....

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

## Module: Mage\_Catalog

Resource: catalog\_product\_custom\_option\_value

**Aliases:** product\_custom\_option\_value

### Methods:

- [product\\_custom\\_option\\_value.add](#) - Add a new custom option value to a selectable custom option
- [product\\_custom\\_option\\_value.list](#) - Retrieve the list of product custom option values
- [product\\_custom\\_option\\_value.info](#) - Retrieve full information about the specified product custom option value
- [product\\_custom\\_option\\_value.update](#) - Update the custom option value
- [product\\_custom\\_option\\_value.remove](#) - Remove the custom option value

### Faults:

| Fault Code | Fault Message                                                           |
|------------|-------------------------------------------------------------------------|
| 101        | Option value with requested id does not exist.                          |
| 102        | Error while adding an option value. Details are in the error message.   |
| 103        | Option with requested id does not exist.                                |
| 104        | Invalid option type.                                                    |
| 105        | Store with requested code/id does not exist.                            |
| 106        | Can not delete option.                                                  |
| 107        | Error while updating an option value. Details are in the error message. |
| 108        | Title field is required.                                                |

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

---

109

Option should have at least one value. Can not delete last value.

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Value Add

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ^

[product\\_custom\\_option\\_value....](#)  
[product\\_custom\\_option\\_value.i...](#)  
[product\\_custom\\_option\\_value.list](#)  
[product\\_custom\\_option\\_value....](#)  
[product\\_custom\\_option\\_value....](#)

[Catalog Product Downloadable](#)  
Link ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

## Module: Complex Product API

Resource: `product_custom_option_value`

### Method:

- `product_custom_option_value.add` (SOAP V1)
- `catalogProductCustomOptionValueAdd` (SOAP V2)

Allows you to add a new custom option value to a custom option. Note that the custom option value can be added only to the option with the **Select** Input Type.

### Arguments:

| Type   | Name      | Description                                 |
|--------|-----------|---------------------------------------------|
| string | sessionId | Session ID                                  |
| string | optionId  | Option ID                                   |
| array  | data      | Array of catalogProductCustomOptionValueAdd |
| string | store     | Store view ID or code (optional)            |

### Return:

| Type    | Description                              |
|---------|------------------------------------------|
| boolean | True if the custom option value is added |

The **catalogProductCustomOptionValueAdd** content is as follows:

| Type   | Name  | Description               |
|--------|-------|---------------------------|
| string | title | Custom option value title |

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

|        |            |                                                                                                   |
|--------|------------|---------------------------------------------------------------------------------------------------|
| string | price      | Custom option value price                                                                         |
| string | price_type | Type of the custom option value price. Can have one of the following values: "fixed" or "percent" |
| string | sku        | Custom option value row SKU                                                                       |
| string | sort_order | Custom option value sort order                                                                    |

#### Faults:

| Fault Code | Fault Message                                                         |
|------------|-----------------------------------------------------------------------|
| 101        | Option value with requested id does not exist.                        |
| 102        | Error while adding an option value. Details are in the error message. |
| 104        | Invalid option type.                                                  |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$productOptionId = 1;// Existing product option ID

// Add custom option value
$customOptionValue = array(
 "title" => "Some value text 1",
 "price" => 10.00,
 "price_type" => "fixed",
 "sku" => "custom_text_option_sku",
 "sort_order" => 0
);
$resultCustomOptionValueAdd = $proxy->call(
 $sessionId,
 "product_custom_option_value.add",
 array(
 $productOptionId,
 array($customOptionValue)
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->catalogProductCustomOptionValueAdd($sessionId, '10', array(array(
'title' => 'value',
'price' => '99.99',
'price_type' => 'fixed',
'sku' => 'sku',
'sort_order' => '1'
)));
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductCustomOptionValueAdd((object)array('sessionId' =>
$sessionId->result, 'optionId' => '10', 'data' => array(array(
'title' => 'value',
'price' => '99.99',
'price_type' => 'fixed',
'sku' => 'sku',
'sort_order' => '1'
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Value Info

[Edit this page on GitHub](#) 

## Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)   
[Value](#) 

product\_custom\_option\_value....  
product\_custom\_option\_value.i...  
product\_custom\_option\_value.list  
product\_custom\_option\_value....  
product\_custom\_option\_value....

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

[Directory](#) 

## Module: Complex Product API

### Resource: product\_custom\_option\_value

#### Method:

- product\_custom\_option\_value.info (SOAP V1)
- catalogProductCustomOptionValueInfo (SOAP V2)

Allows you to retrieve full information about the specified product custom option value.

#### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | valueId   | Value ID                         |
| string | store     | Store view ID or code (optional) |

#### Return:

| Type  | Name   | Description                                        |
|-------|--------|----------------------------------------------------|
| array | result | Array of catalogProductCustomOptionValueInfoEntity |

The **catalogProductCustomOptionValueInfoEntity** content is as follows:

| Type   | Name      | Description                 |
|--------|-----------|-----------------------------|
| string | value_id  | Option value ID             |
| string | option_id | Option ID                   |
| string | sku       | Custom option value row SKU |

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

|        |                    |                                                                          |
|--------|--------------------|--------------------------------------------------------------------------|
| string | sort_order         | Option value sort order                                                  |
| string | default_price      | Option value default price                                               |
| string | default_price_type | Default price type. Possible values are as follows: "fixed" or "percent" |
| string | store_price        | Option value store price                                                 |
| string | store_price_type   | Store price type. Possible values are as follows: "fixed" or "percent"   |
| string | price              | Option value price                                                       |
| string | price_type         | Price type. Possible values are as follows: "fixed" or "percent"         |
| string | default_title      | Option value default title                                               |
| string | store_title        | Option value store title                                                 |
| string | title              | Option value title                                                       |

#### Faults:

| Fault Code | Fault Message                                  |
|------------|------------------------------------------------|
| 101        | Option value with requested id does not exist. |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'product_custom_option_value.info', '5');
var_dump($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductCustomOptionValueInfo($sessionId, '5');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductCustomOptionValueInfo((object)array('sessionId' =>
$sessionId->result, 'valueId' => '5'));

var_dump($result->result);
```

## Response Example SOAP V1

```
array
 'option_id' => string '5' (length=1)
 'sku' => string 'slider' (length=6)
 'sort_order' => string '0' (length=1)
 'default_title' => string 'slider' (length=6)
 'store_title' => string 'slider' (length=6)
 'title' => string 'slider' (length=6)
 'default_price' => string '239.9900' (length=8)
 'default_price_type' => string 'fixed' (length=5)
 'store_price' => string '239.9900' (length=8)
 'store_price_type' => string 'fixed' (length=5)
 'price' => string '239.9900' (length=8)
 'price_type' => string 'fixed' (length=5)
 'value_id' => string '2' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Value List

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▲

[product\\_custom\\_option\\_value....](#)

[product\\_custom\\_option\\_value.i...](#)

[product\\_custom\\_option\\_value.list](#)

[product\\_custom\\_option\\_value....](#)

[product\\_custom\\_option\\_value....](#)

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

## Module: Complex Product API

Resource: `product_custom_option_value`

### Method:

- `product_custom_option_value.list` (SOAP V1)
- `catalogProductCustomOptionValueList` (SOAP V2)

Allows you to retrieve the list of product custom option values. Note that the method is available only for the option **Select** Input Type.

### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | optionId  | Option ID                        |
| string | store     | Store view ID or code (optional) |

### Returns:

| Type  | Name   | Description                                  |
|-------|--------|----------------------------------------------|
| array | result | Array of catalogProductCustomOptionValueList |

The **catalogProductCustomOptionValueListEntity** content is as follows:

| Type   | Name     | Description               |
|--------|----------|---------------------------|
| string | value_id | Custom option value ID    |
| string | title    | Custom option value title |

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

|        |            |                                                   |
|--------|------------|---------------------------------------------------|
| string | price      | Option value price                                |
| string | price_type | Price type. Possible values: "fixed" or "percent" |
| string | sku        | Custom option value SKU                           |
| string | sort_order | Option value sort order (optional)                |

#### Faults:

| Fault Code | Fault Message                                                         |
|------------|-----------------------------------------------------------------------|
| 101        | Provided data is invalid.                                             |
| 102        | Error while adding an option value. Details are in the error message. |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'product_custom_option_value.list', '3');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductCustomOptionValueList($sessionId, '3');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductCustomOptionValueList((object)array('sessionId' => $sessionId->result, 'optionId' => '3'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'value_id' => string '1' (length=1)
```

```
'title' => string 'monoblock' (length=9)
'price' => string '139.9900' (length=8)
'price_type' => string 'fixed' (length=5)
'sku' => string 'monoblock' (length=9)
'sort_order' => string '0' (length=1)
1 =>
array
'value_id' => string '2' (length=1)
'title' => string 'slider' (length=6)
'price' => string '239.9900' (length=8)
'price_type' => string 'fixed' (length=5)
'sku' => string 'slider' (length=6)
'sort_order' => string '0' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Value Remove

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom Option](#) ▼

[Catalog Product Custom Option Value](#) ▼

product\_custom\_option\_value....  
product\_custom\_option\_value.i...  
product\_custom\_option\_value.list  
[product\\_custom\\_option\\_value](#)....  
product\_custom\_option\_value....

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

## Module: Complex Product API

Resource: product\_custom\_option\_value

### Method:

- product\_custom\_option\_value.remove (SOAP V1)
- catalogProductCustomOptionValueRemove (SOAP V2)

Allows you to remove the custom option value from a product.

### Arguments:

| Type   | Name      | Description            |
|--------|-----------|------------------------|
| string | sessionId | Session ID             |
| string | valueId   | Custom option value ID |

### Return:

| Type        | Description                                    |
|-------------|------------------------------------------------|
| boolean\int | True (1) if the custom option value is removed |

### Faults:

| Fault Code | Fault Message                                                     |
|------------|-------------------------------------------------------------------|
| 103        | Option with requested id does not exist.                          |
| 106        | Can not delete option.                                            |
| 109        | Option should have at least one value. Can not delete last value. |

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$productOptionId = 4;// Existing option ID
// Get last value from option values list
$optionValues = $proxy->call($sessionId, "product_custom_option_value.list",
array($productOptionId));
$optionValue = reset($optionValues);
$valueId = $optionValue['value_id'];

$result = $proxy->call(
 $sessionId,
 "product_custom_option_value.remove",
 array(
 $valueId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductCustomOptionValueRemove($sessionId, '4');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductCustomOptionValueRemove((object)array('sessionId' =>
$sessionId->result, 'valueId' => '4'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Custom Value Update

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom Option](#) ▼

[Catalog Product Custom Option Value](#) ▼

product\_custom\_option\_value....  
product\_custom\_option\_value.i...  
product\_custom\_option\_value.list  
product\_custom\_option\_value....  
product\_custom\_option\_value....

[Catalog Product Downloadable Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

## Module: Complex Product API

Resource: product\_custom\_option\_value

### Method:

- product\_custom\_option\_value.update (SOAP V1)
- catalogProductCustomOptionValueUpdate (SOAP V2)

Allows you to update the product custom option value.

### Arguments:

| Type   | Name      | Description                                          |
|--------|-----------|------------------------------------------------------|
| string | sessionId | Session ID                                           |
| string | valueId   | Value ID                                             |
| array  | data      | Array of catalogProductCustomOptionValueUpdateEntity |
| string | storeId   | Store view ID or code (optional)                     |

### Return:

| Type    | Description                                |
|---------|--------------------------------------------|
| boolean | True if the custom option value is updated |

The **catalogProductCustomOptionValueUpdateEntity** content is as follows:

| Type   | Name  | Description        |
|--------|-------|--------------------|
| string | title | Option value title |
| string | price | Option value price |

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

|        |            |                                                                  |
|--------|------------|------------------------------------------------------------------|
| string | price_type | Price type. Possible values are as follows: "fixed" or "percent" |
| string | sku        | Custom option value row SKU                                      |
| string | sort_order | Custom option value sort order                                   |

#### Faults:

| Fault Code | Fault Message                                                           |
|------------|-------------------------------------------------------------------------|
| 101        | Option value with requested id does not exist.                          |
| 103        | Option with requested id does not exist.                                |
| 104        | Invalid option type.                                                    |
| 107        | Error while updating an option value. Details are in the error message. |
| 108        | Title field is required.                                                |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$productOptionId = 1;// Existing option ID

// Get last value from option values list
$optionValues = $proxy->call($sessionId, "product_custom_option_value.list",
array($productOptionId));
$optionValue = reset($optionValues);
$valueId = $optionValue['value_id'];
// Update custom option value
$customOptionValue = array(
 "title" => "new title",
 "price" => 12.00,
 "price_type" => "percent",
 "sku" => "custom_text_option_2",
 "sort_order" => 2
);
$resultCustomOptionValueUpdate = $proxy->call(
 $sessionId,
 "product_custom_option_value.update",
 array(
 $valueId,
 $customOptionValue
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->catalogProductCustomOptionValueUpdate($sessionId, '2', array(
'title' => 'value',
'price' => '20',
```

```
'price_type' => 'fixed',
'sku' => 'sku'
)) ;

var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductCustomOptionValueUpdate((object)array('sessionId' =>
$sessionId->result, 'valueId' => '2', 'data' => ((object)array(
'title' => 'value',
'sku' => 'sku',
'price' => '199',
'price_type' => 'percent'
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product Downloadable Link

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

### Link ^

[product\\_downloadable\\_link.add](#)

[product\\_downloadable\\_link.list](#)

[product\\_downloadable\\_link.re...](#)

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

[Sales](#) ▼

## Module: Complex Product API

Resource: product\_downloadable\_link

### Methods:

- [product\\_downloadable\\_link.add](#) - Add a new link to the downloadable product
- [product\\_downloadable\\_link.list](#) - Get the list of links for a downloadable product
- [product\\_downloadable\\_link.remove](#) - Remove a link from a downloadable product

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

---

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Downloadable Link Add

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ^

[product\\_downloadable\\_link.add](#)

[product\\_downloadable\\_link.list](#)

[product\\_downloadable\\_link.re...](#)

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

### Catalog Inventory

### Checkout

### Create Your Own API

### Customer

### Directory

### Sales

## Module: Complex Product API

Resource: product\_downloadable\_link

### Method:

- `product_downloadable_link.add` (SOAP V1)
- `catalogProductDownloadableLinkAdd` (SOAP V2)

Allows you to add a new link to a downloadable product.

### Arguments:

| Type   | Name           | Description                                                                          |
|--------|----------------|--------------------------------------------------------------------------------------|
| string | sessionId      | Session ID                                                                           |
| string | productId      | Product ID                                                                           |
| array  | resource       | Array of catalogProductDownloadableLinkAddEntity                                     |
| string | resourceType   | Resource type. Can have one of the following values: "sample" or "link".             |
| string | store          | Store view ID or code (optional)                                                     |
| string | identifierType | Type of the product identifier. Can have one of the following values: "sku" or "id". |

### Return:

| Type | Name   | Description                                         |
|------|--------|-----------------------------------------------------|
| int  | result | Result of adding a link to the downloadable product |

The `catalogProductDownloadableLinkAddEntity` content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

**Enterprise Customer Balance** ▾**Enterprise Gift Card** ▾**Enterprise Gift Message** ▾**Miscellaneous** ▾**WS-I Compliance**

|        |                     |                                                                                |
|--------|---------------------|--------------------------------------------------------------------------------|
| string | title               | Link title                                                                     |
| string | price               | Custom option value row price                                                  |
| int    | is_unlimited        | Defines whether the number of downloads is unlimited                           |
| int    | number_of_downloads | Maximum number of possible downloads                                           |
| int    | is_shareable        | Defines whether the link is shareable                                          |
| array  | sample              | Array of catalogProductDownloadableLinkAddSampleEntity                         |
| string | type                | Type of the data source. Can have one of the following values: "file" or "url" |
| array  | file                | Array of catalogProductDownloadableLinkFileEntity                              |
| string | link_url            | Link URL address                                                               |
| string | sample_url          | Sample URL address                                                             |
| int    | sort_order          | Link sort order                                                                |

The **catalogProductDownloadableLinkAddSampleEntity** content is as follows:

| Type   | Name | Description                                                                    |
|--------|------|--------------------------------------------------------------------------------|
| string | type | Type of the data source. Can have one of the following values: "file" or "url" |
| array  | file | Array of catalogProductDownloadableLinkFileEntity                              |
| string | url  | URL to upload                                                                  |

The **catalogProductDownloadableLinkFileEntity** content is as follows:

| Type   | Name           | Description         |
|--------|----------------|---------------------|
| string | name           | File name           |
| string | base64_content | BASE64 encoded file |

**Faults:**

| Fault Code | Fault Message                                    |
|------------|--------------------------------------------------|
| 414        | Unable to save action. Details in error message. |
| 415        | Validation error has occurred.                   |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$filePath = '/var/www/ws/tests/WebService/etc/Modules/Downloadable/Product/Link';
$downloadableProductId = 'downloadable_demo_product';

$items = array(
 'small' => array(
 'link' => array(
```

```

 'title' => 'Test file',
 'price' => '123',
 'is_unlimited' => '1',
 'number_of_downloads' => '111',
 'is_shareable' => '0',
 'sample' => array(
 'type' => 'file',
 'file' =>
 array(
 'filename' => 'files/test.txt',
),
 'url' => 'http://www.magentocommerce.com/img/logo.gif',
),
 'type' => 'file',
 'file' =>
 array(
 'filename' => 'files/test.txt',
),
 'link_url' => 'http://www.magentocommerce.com/img/logo.gif',
),
 'sample' => array(
 'title' => 'Test sample file',
 'type' => 'file',
 'file' => array(
 'filename' => 'files/image.jpg',
),
 'sample_url' => 'http://www.magentocommerce.com/img/logo.gif',
 'sort_order' => '3',
)
),
'big' => array(
 'link' => array(
 'title' => 'Test url',
 'price' => '123',
 'is_unlimited' => '0',
 'number_of_downloads' => '111',
 'is_shareable' => '1',
 'sample' => array(
 'type' => 'url',
 'file' => array(
 'filename' => 'files/book.pdf',
),
 'url' => 'http://www.magentocommerce.com/img/logo.gif',
),
 'type' => 'url',
 'file' => array(
 'filename' => 'files/song.mp3',
),
 'link_url' => 'http://www.magentocommerce.com/img/logo.gif',
),
 'sample' => array(
 'title' => 'Test sample url',
 'type' => 'url',
 'file' => array(
 'filename' => 'files/image.jpg',
),
 'sample_url' => 'http://www.magentocommerce.com/img/logo.gif',
 'sort_order' => '3',
)
)
);
}

$result = true;

```

```

foreach ($items as $item) {
 foreach ($item as $key => $value) {
 if ($value['type'] == 'file') {
 $filePath = $filesPath . '/' . $value['file']['filename'];
 $value['file'] = array('name' => str_replace('/', '_', $value['file']['filename']), 'base64_content' => base64_encode(file_get_contents($filePath)), 'type' => $value['type']);
 }
 if ($value['sample']['type'] == 'file') {
 $filePath = $filesPath . '/' . $value['sample']['file']['filename'];
 $value['sample']['file'] = array('name' => str_replace('/', '_', $value['sample']['file']['filename']), 'base64_content' => base64_encode(file_get_contents($filePath)));
 }
 if (!$proxy->call(
 $sessionId,
 'product_downloadable_link.add',
 array($downloadableProductId, $value, $key)
))
 {
 $result = false;
 }
 }
}

```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductDownloadableLinkAdd((object)array('sessionId' => $sessionId->result, 'productId' => '3', 'resourceType' => 'link', 'resource' => ((object)array(
 'title' => 'link',
 'price' => '10.99',
 'sample' => array(
 'type' => 'url',
 'url' => 'http://sometesturl.com'
))));
var_dump($result->result);

```

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductDownloadableLinkAdd((object)array('sessionId' => $sessionId->result, 'productId' => '3', 'resourceType' => 'link', 'resource' => ((object)array(
 'title' => 'link_2',
 'price' => '11.99',
 'type' => 'file',
 'file' => array(
 'name' => 'file_test',
 'base64_content' => '/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwyHBwcGBwcICQsJCAg
)
))));
```

```
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Downloadable Link List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)  
[Link](#) ^

[product\\_downloadable\\_link.add](#)

[product\\_downloadable\\_link.list](#)

[product\\_downloadable\\_link.re...](#)

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

[Sales](#) ▼

### Module: Complex Product API

Resource: `product_downloadable_link`

#### Method:

- `product_downloadable_link.list` (SOAP V1)
- `catalogProductDownloadableLinkList` (SOAP V2)

Allows you to retrieve a list of links of a downloadable product.

#### Arguments:

| Type   | Name           | Description                                                    |
|--------|----------------|----------------------------------------------------------------|
| string | sessionId      | Session ID                                                     |
| string | productId      | Product ID or SKU                                              |
| string | store          | Store view ID or code (optional)                               |
| string | identifierType | Defines whether the product ID or SKU is passed in the request |

#### Return:

| Type  | Name   | Description                                       |
|-------|--------|---------------------------------------------------|
| array | result | Array of catalogProductDownloadableLinkListEntity |

The `catalogProductDownloadableLinkListEntity` content is as follows:

| Type  | Name    | Description                                         |
|-------|---------|-----------------------------------------------------|
| array | links   | Array of catalogProductDownloadableLinkEntity       |
| array | samples | Array of catalogProductDownloadableLinkSampleEntity |

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

The **catalogProductDownloadableLinkEntity** content is as follows:

| Type   | Name                | Description                                                                         |
|--------|---------------------|-------------------------------------------------------------------------------------|
| string | link_id             | Link ID                                                                             |
| string | title               | Link title                                                                          |
| string | price               | Downloadable link price value                                                       |
| int    | number_of_downloads | Maximum number of possible downloads                                                |
| int    | is_unlimited        | Defines whether the number of downloads is unlimited                                |
| int    | is_shareable        | Defines whether the link is shareable                                               |
| string | link_url            | Link URL address                                                                    |
| string | link_type           | Type of the link data source. Can have one of the following values: "file" or "url" |
| string | sample_file         | Sample file name                                                                    |
| string | sample_url          | Sample URL                                                                          |
| string | sample_type         | Type of sample data source. Can have one of the following values: "file" or "url"   |
| int    | sort_order          | Link sort order                                                                     |
| array  | file_save           | Array of catalogProductDownloadableLinkFileInfoEntity                               |
| array  | sample_file_save    | Array of catalogProductDownloadableLinkFileInfoEntity                               |

The **catalogProductDownloadableLinkSampleEntity** content is as follows:

| Type   | Name          | Description                                                        |
|--------|---------------|--------------------------------------------------------------------|
| string | sample_id     | Sample ID                                                          |
| string | product_id    | Product ID                                                         |
| string | sample_file   | Sample file name                                                   |
| string | sample_url    | Sample URL                                                         |
| string | sample_type   | Sample type. Can have one of the following values: "file" or "url" |
| string | sort_order    | Sort order                                                         |
| string | default_title | Default title                                                      |
| string | store_title   | Store title                                                        |
| string | title         | Sample title                                                       |

The **catalogProductDownloadableLinkFileInfoEntity** content is as follows:

| Type   | Name | Description |
|--------|------|-------------|
| string | file | File        |
| string | name | File name   |

| int    | size   | File size |
|--------|--------|-----------|
| string | status | Status    |

### Faults:

No Faults

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$downloadableProductId = '5';

$resultList = $proxy->call(
 $sessionId,
 'product_downloadable_link.list',
 array($downloadableProductId)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductDownloadableLinkList($sessionId, '5', null, 'sku');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductDownloadableLinkList((object)array('sessionId' => $sessionId->result, 'productId' => '5'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
 'links' =>
 array
 0 =>
 array
 'link_id' => string '1' (length=1)
 'title' => string 'link 1' (length=11)
 'price' => string '30.0000' (length=7)
 'number_of_downloads' => string '0' (length=1)
 'is_shareable' => string '1' (length=1)
 'link_url' => null
 'link_type' => string 'file' (length=4)
 'sample_file' => string '/s/o/software.jpg' (length=17)
 'sample_url' => null
 'sample_type' => string 'file' (length=4)
```

```
'sort_order' => string '1' (length=1)
'file_save' =>
 array
 ...
'sample_file_save' =>
 array
 ...
'is_unlimited' => int 1
'samples' =>
 array
 0 =>
 array
 'sample_id' => string '1' (length=1)
 'product_id' => string '5' (length=1)
 'sample_url' => null
 'sample_file' => string '/s/o/software.jpg' (length=17)
 'sample_type' => string 'file' (length=4)
 'sort_order' => string '2' (length=1)
 'default_title' => string 'Sample 1' (length=8)
 'store_title' => string 'Sample 1' (length=8)
 'title' => string 'Sample 1' (length=8)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Downloadable Link Remove

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▲

[product\\_downloadable\\_link.add](#)

[product\\_downloadable\\_link.list](#)

[product\\_downloadable\\_link.re...](#)

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

## Catalog Inventory ▼

## Checkout ▼

## Create Your Own API

## Customer ▼

## Directory ▼

## Sales ▼

## Module: Complex Product API

Resource: product\_downloadable\_link

### Method:

- `product_downloadable_link.remove` (SOAP V1)
- `catalogProductDownloadableLinkRemove` (SOAP V2)

Allows you to remove a link/sample from a downloadable product.

### Arguments:

| Type   | Name            | Description                                                             |
|--------|-----------------|-------------------------------------------------------------------------|
| string | sessionId       | Session ID                                                              |
| string | linkId/sampleId | Link/sample ID                                                          |
| string | resourceType    | Resource type. Can have one of the following values: 'sample' or 'link' |

### Return:

| Type    | Description                                                    |
|---------|----------------------------------------------------------------|
| boolean | True if the link/sample is removed from a downloadable product |

### Faults:

| Fault Code | Fault Message                                    |
|------------|--------------------------------------------------|
| 412        | Link or sample with specified ID was not found.  |
| 415        | Validation error has occurred.                   |
| 416        | Unable to remove link. Details in error message. |

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$sampleId = 7;
$linkId = 9;

$resultSample = $proxy->call(
 $sessionId,
 'product_downloadable_link.remove',
 array($sampleId, 'sample')
);

$resultLink = $proxy->call(
 $sessionId,
 'product_downloadable_link.remove',
 array($linkId, 'link')
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductDownloadableLinkRemove($sessionId, '7', 'sample');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductDownloadableLinkRemove((object)array('sessionId' => $sessionId->result, 'linkId' => '7', 'resourceType' => 'sample'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product Link

[Edit this page on GitHub](#) 

## Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

`catalog_product_link.assign`

`catalog_product_link.attributes`

`catalog_product_link.list`

`catalog_product_link.remove`

`catalog_product_link.types`

`catalog_product_link.update`

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Product Links

Allows you to manage links for products, including related, cross-sells, up-sells, and grouped.

**Resource Name:** catalog\_product\_link

#### Aliases:

- product\_link

#### Methods:

- [catalog\\_product\\_link.list](#) - Retrieve products linked to the specified product
- [catalog\\_product\\_link.assign](#) - Link a product to another product
- [catalog\\_product\\_link.update](#) - Update a product link
- [catalog\\_product\\_link.remove](#) - Remove a product link
- [catalog\\_product\\_link.types](#) - Retrieve product link types
- [catalog\\_product\\_link.attributes](#) - Retrieve product link type attributes

### Faults

| Fault Code | Fault Message                                 |
|------------|-----------------------------------------------|
| 100        | Given invalid link type.                      |
| 101        | Product not exists.                           |
| 102        | Invalid data given. Details in error message. |
| 104        | Product link not removed.                     |

## Examples

### Example 1. Working with product links

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

// Get list of related products
var_dump($proxy->call($sessionId, 'product_link.list', array('related', 'Sku')));

// Assign related product
$proxy->call($sessionId, 'product_link.assign', array('related', 'Sku', 'Sku2',
array('position'=>0, 'qty'=>56)));

var_dump($proxy->call($sessionId, 'product_link.list', array('related', 'Sku')));

// Update related product
$proxy->call($sessionId, 'product_link.update', array('related', 'Sku', 'Sku2',
array('position'=>2)));

var_dump($proxy->call($sessionId, 'product_link.list', array('related', 'Sku')));

// Remove related product
$proxy->call($sessionId, 'product_link.remove', array('related', 'Sku', 'Sku2'));

var_dump($proxy->call($sessionId, 'product_link.list', array('related', 'Sku')));
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Link Assign

[Edit this page on GitHub](#) 

### Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[catalog\\_product\\_link.assign](#)

[catalog\\_product\\_link.attributes](#)

[catalog\\_product\\_link.list](#)

[catalog\\_product\\_link.remove](#)

[catalog\\_product\\_link.types](#)

[catalog\\_product\\_link.update](#)

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

## Module: Mage\_Catalog

Resource: catalog\_product\_link

### Aliases:

- product\_link

### Method:

- catalog\_product\_link.assign (SOAP V1)
- catalogProductLinkAssign (SOAP V2)

Allows you to assign a product link (cross\_sell, grouped, related, or up\_sell) to another product.

### Aliases:

- product\_link.assign

### Arguments:

| Type   | Name                          | Description                                                                |
|--------|-------------------------------|----------------------------------------------------------------------------|
| string | sessionId                     | Session ID                                                                 |
| string | type                          | Type of the link (cross_sell, grouped, related, or up_sell)                |
| string | product\productId             | Product ID or SKU                                                          |
| string | linkedProduct\linkedProductId | Product ID or SKU for the link                                             |
| array  | data                          | Array of catalogProductLinkEntity                                          |
| string | identifierType                | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

| Type    | Description                                 |
|---------|---------------------------------------------|
| boolean | True if the link is assigned to the product |

The **catalogProductLinkEntity** content is as follows:

| Type   | Name       | Description                                                 |
|--------|------------|-------------------------------------------------------------|
| string | product_id | Product ID                                                  |
| string | type       | Type of the link (cross_sell, grouped, related, or up_sell) |
| string | set        | Product attribute set                                       |
| string | sku        | Product SKU                                                 |
| string | position   | Position of the product                                     |
| string | qty        | Quantity of products                                        |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apikey');

$result = $client->call($session, 'catalog_product_link.assign', array('type' =>
'related', 'product' => '1', 'linkedProduct' => '4'));
var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductLinkAssign($sessionId, 'related', '1', '4');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductLinkAssign((object)array('sessionId' => $sessionId-
>result, 'type' => 'related', 'productId' => '1', 'linkedProductId' => '4'));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Link Attributes

[Edit this page on GitHub](#) 

### Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[catalog\\_product\\_link.assign](#)

[catalog\\_product\\_link.attributes](#)

[catalog\\_product\\_link.list](#)

[catalog\\_product\\_link.remove](#)

[catalog\\_product\\_link.types](#)

[catalog\\_product\\_link.update](#)

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

## Module: Mage\_Catalog

Resource: catalog\_product\_link

### Aliases:

- product\_link

### Method:

- catalog\_product\_link.attributes (SOAP V1)
- catalogProductLinkAttributes (SOAP V2)

Allows you to retrieve the product link type attributes.

### Aliases:

- product\_link.attributes

### Arguments:

| Type   | Name      | Description                                                 |
|--------|-----------|-------------------------------------------------------------|
| string | sessionId | Session ID                                                  |
| string | type      | Type of the link (cross_sell, up_sell, related, or grouped) |

### Returns:

| Type  | Name   | Description                                |
|-------|--------|--------------------------------------------|
| array | result | Array of catalogProductLinkAttributeEntity |

The **catalogProductLinkAttributeEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|                             |   |
|-----------------------------|---|
| Directory                   | ✓ |
| Sales                       | ✓ |
| Enterprise Customer Balance | ✓ |
| Enterprise Gift Card        | ✓ |
| Enterprise Gift Message     | ✓ |
| Miscellaneous               | ✓ |
| WS-I Compliance             |   |

|        |      |                |
|--------|------|----------------|
| string | code | Attribute code |
| string | type | Attribute type |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_link.attributes', 'related');
var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductLinkAttributes($sessionId, 'related');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductLinkAttributes((object)array('sessionId' => $sessionId->result, 'type' => 'related'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'code' => string 'position' (length=8)
 'type' => string 'int' (length=3)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Link List

[Edit this page on GitHub](#) 

### Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[catalog\\_product\\_link.assign](#)

[catalog\\_product\\_link.attributes](#)

[catalog\\_product\\_link.list](#)

[catalog\\_product\\_link.remove](#)

[catalog\\_product\\_link.types](#)

[catalog\\_product\\_link.update](#)

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

## Module: Mage\_Catalog

Resource: catalog\_product\_link

### Aliases:

- product\_link

### Method:

- catalog\_product\_link.list (SOAP V1)
- catalogProductLinkList (SOAP V2)

Allows you to retrieve the list of linked products for a specific product.

### Aliases:

- product\_link.list

### Arguments:

| Type   | Name              | Description                                                                |
|--------|-------------------|----------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                 |
| string | type              | Type of the link (cross_sell, up_sell, related, or grouped)                |
| string | product\productId | Product ID or SKU                                                          |
| string | identifierType    | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

| Type  | Name   | Description                       |
|-------|--------|-----------------------------------|
| array | result | Array of catalogProductLinkEntity |

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

The **catalogProductLinkEntity** content is as follows:

| Type   | Name       | Description           |
|--------|------------|-----------------------|
| string | product_id | Product ID            |
| string | type       | Type of the link      |
| string | set        | Product attribute set |
| string | sku        | Product SKU           |
| string | position   | Position              |
| string | qty        | Quantity              |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_link.list', array('type' =>
'related', 'product' => '1'));
var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductLinkList($sessionId, 'related', '1');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductLinkList((object)array('sessionId' => $sessionId-
>result, 'type' => 'related', 'productId' => '1'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
'product_id' => string '3' (length=1)
```

```
'type' => string 'simple' (length=6)
'set' => string '4' (length=1)
'sku' => string 'canonxt' (length=7)
'position' => string '1' (length=1)
1 =>
array
 'product_id' => string '4' (length=1)
 'type' => string 'simple' (length=6)
 'set' => string '4' (length=1)
 'sku' => string 'canon_powershot' (length=15)
 'position' => string '0' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Link Remove

[Edit this page on GitHub](#) 

### Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[catalog\\_product\\_link.assign](#)

[catalog\\_product\\_link.attributes](#)

[catalog\\_product\\_link.list](#)

[catalog\\_product\\_link.remove](#)

[catalog\\_product\\_link.types](#)

[catalog\\_product\\_link.update](#)

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

## Module: Mage\_Catalog

Resource: catalog\_product\_link

### Aliases:

- product\_link

### Method:

- catalog\_product\_link.remove (SOAP V1)
- catalogProductLinkRemove (SOAP V2)

Allows you to remove the product link from a specific product.

### Aliases:

- product\_link.remove

### Arguments:

| Type   | Name                          | Description                                                                |
|--------|-------------------------------|----------------------------------------------------------------------------|
| string | sessionId                     | Session ID                                                                 |
| string | type                          | Type of the link (cross_sell, up_sell, related, or grouped)                |
| string | product\productId             | Product ID or SKU                                                          |
| string | linkedProduct\linkedProductId | Product ID or SKU for the link                                             |
| string | identifierType                | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

| Type | Description |
|------|-------------|
|      |             |

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

boolean\int

True (1) if the link is removed from a product

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_link.remove', array('type' =>
'related', 'product' => '1', 'linkedProduct' => '4'));
var_dump ($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if
necessary

$result = $proxy->catalogProductLinkRemove($sessionId, 'related', '1', '4');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductLinkRemove((object)array('sessionId' => $sessionId-
>result, 'type' => 'related', 'productId' => '1', 'linkedProductId' => '4'));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Link Types

[Edit this page on GitHub](#) 

## Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[catalog\\_product\\_link.assign](#)

[catalog\\_product\\_link.attributes](#)

[catalog\\_product\\_link.list](#)

[catalog\\_product\\_link.remove](#)

[catalog\\_product\\_link.types](#)

[catalog\\_product\\_link.update](#)

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

## Module: Mage\_Catalog

Resource: catalog\_product\_link

### Aliases:

- product\_link

### Method:

- catalog\_product\_link.types (SOAP V1)
- catalogProductLinkTypes (SOAP V2)

Allows you to retrieve the list of product link types.

### Aliases:

- product\_link.types

### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

### Returns:

| Type          | Name   | Description         |
|---------------|--------|---------------------|
| ArrayOfString | result | Array of link types |

## Examples

### Request Example SOAP V1

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

---

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'product_link.types');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductLinkTypes($sessionId);
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductLinkTypes((object)array('sessionId' => $sessionId->result));

var_dump($result->result);
```

## Response Example SOAP V1

```
array
 0 => string 'related' (length=7)
 1 => string 'up_sell' (length=7)
 2 => string 'cross_sell' (length=10)
 3 => string 'grouped' (length=7)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Link Update

[Edit this page on GitHub](#) 

## Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[catalog\\_product\\_link.assign](#)

[catalog\\_product\\_link.attributes](#)

[catalog\\_product\\_link.list](#)

[catalog\\_product\\_link.remove](#)

[catalog\\_product\\_link.types](#)

[catalog\\_product\\_link.update](#)

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

## Module: Mage\_Catalog

Resource: catalog\_product\_link

### Aliases:

- product\_link

### Method:

- catalog\_product\_link.update (SOAP V1)
- catalogProductLinkUpdate (SOAP V2)

Allows you to update the product link.

### Aliases:

- product\_link.update

### Arguments:

| Type   | Name                          | Description                                                                |
|--------|-------------------------------|----------------------------------------------------------------------------|
| string | sessionId                     | Session ID                                                                 |
| string | type                          | Type of the link (cross_sell, grouped, related, or up_sell)                |
| string | product\productId             | Product ID or SKU                                                          |
| string | linkedProduct\linkedProductId | Product ID or SKU for the link                                             |
| array  | data                          | Array of catalogProductLinkEntity                                          |
| string | identifierType                | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

| Type        | Name   | Description                     |
|-------------|--------|---------------------------------|
| boolean/int | result | True (1) if the link is updated |

The **catalogProductLinkEntity** content is as follows:

| Type   | Name       | Description           |
|--------|------------|-----------------------|
| string | product_id | Product ID            |
| string | type       | Type of the link      |
| string | set        | Product attribute set |
| string | sku        | Product SKU           |
| string | position   | Position              |
| string | qty        | Quantity              |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$productId = '1';
$linkedProductId = '2';
$data = array(
 'position' => '50'
);

$result = $proxy->call(
 $sessionId,
 'catalog_product_link.update',
 array(
 'cross_sell',
 $productId,
 $linkedProductId,
 $data
)
);

```

### Request Example SOAP V2

```
<?php

$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$session = $client->login('apiUser', 'apiKey');

$data = array(
 "position" => 15
);

$identifierType = "product_id";
$type = "related";
$product = "1";
$linkedinProduct = "3";
```

```
$orders = $client->catalogProductLinkUpdate($session, $type, $product, $linkedProduct,
$data, $identifierType);

echo 'Number of results: ' . count($orders) . '
';
var_dump ($orders);
?>
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductLinkUpdate((object)array('sessionId' => $sessionId-
>result, 'type' => 'cross_sell', 'productId' => '1', 'linkedProductId' => '2', 'data'
=> ((object)array(
'position' => '1'
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product Tag

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ^

[product\\_tag.add](#)

[product\\_tag.info](#)

[product\\_tag.list](#)

[product\\_tag.remove](#)

[product\\_tag.update](#)

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

## Module: Tag

Allows you to operate with product tags.

Resource: catalog\_product\_tag

**Aliases:** product\_tag

### Methods:

- [product\\_tag.list](#) - Retrieve the list of tags by the product ID
- [product\\_tag.info](#) - Retrieve information about a product tag
- [product\\_tag.add](#) - Add one or more tags to a product
- [product\\_tag.update](#) - Update an existing product tag
- [product\\_tag.remove](#) - Remove a product tag

### Faults:

| Fault Code | Fault Message                                       |
|------------|-----------------------------------------------------|
| 101        | Requested store does not exist.                     |
| 102        | Requested product does not exist.                   |
| 103        | Requested customer does not exist.                  |
| 104        | Requested tag does not exist.                       |
| 105        | Provided data is invalid.                           |
| 106        | Error while saving tag. Details in error message.   |
| 107        | Error while removing tag. Details in error message. |



Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

## Example:

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$productID = 2;
$customerId = 10002;
$store = 'default';

// Add tags
$data = array('product_id' => $productId, 'store' => $store, 'customer_id' =>
$customerId, 'tag' => "First 'Second tag' Third");
echo "Adding Tag... ";
$addResult = $proxy->call(
 $sessionId,
 "product_tag.add",
 array($data)
);
echo ((count($addResult) == 3) ? "Done!" : "Fail!");
echo "
";
print_r($addResult);
$tagId = reset($addResult);

// Get tag info
echo "
Get Tag Info (id = $tagId)... ";
$infoResult = $proxy->call(
 $sessionId,
 "product_tag.info",
 array($tagId, $store)
);
echo "Done!
";
print_r($infoResult);

// Update tag data
$data = array('status' => -1, 'base_popularity' => 12, 'name' => 'Changed name');
echo "
Update Tag (id = $tagId)... ";
$updateResult = $proxy->call(
 $sessionId,
 "product_tag.update",
 array($tagId, $data, $store)
);
echo ($updateResult ? "Done!" : "Fail!");

// Retrieve list of tags by product
echo "
Tag list for product with id = $productId... ";
$listResult = $proxy->call(
 $sessionId,
 "product_tag.list",
 array($productId, $store)
);
echo (count($listResult) ? "Done!" : "Fail!");
echo "
";
print_r($listResult);

// Remove existing tag
echo "
Remove Tag (id = $tagId)... ";
$removeResult = $proxy->call(
 $sessionId,
 "product_tag.remove",
 array($tagId)
);
echo ($removeResult ? "Done!" : "Fail!");
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Tag Add

[Edit this page on GitHub](#) 

### Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[product\\_tag.add](#)

[product\\_tag.info](#)

[product\\_tag.list](#)

[product\\_tag.remove](#)

[product\\_tag.update](#)

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

### Catalog Inventory ^

### Checkout ^

### Create Your Own API

### Customer ^

### Directory ^

## Module: Tag Api

Resource: catalog\_product\_tag

**Aliases:** product\_tag

### Method:

- catalog\_product\_tag.add (SOAP V1)
- catalogProductTagAdd (SOAP V2)

Allows you to add one or more tags to a product.

### Arguments:

| Type   | Name      | Description                         |
|--------|-----------|-------------------------------------|
| string | sessionId | Session ID                          |
| array  | data      | Array of catalogProductTagAddEntity |

### Return:

| Type  | Name   | Description                                                                          |
|-------|--------|--------------------------------------------------------------------------------------|
| array | result | Associative array of added tags with the tag name as a key and the tag ID as a value |

The **catalogProductTagAddEntity** content is as follows:

| Type   | Name       | Description                                                                                                                                    |
|--------|------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| string | tag        | Tag to be added (can contain several tags separated with white spaces). A tag that contains several words should be enclosed in single quotes. |
| string | product_id | Product ID                                                                                                                                     |

Sales ✓

Enterprise Customer Balance ✓

Enterprise Gift Card ✓

Enterprise Gift Message ✓

Miscellaneous ✓

WS-I Compliance

|        |             |             |
|--------|-------------|-------------|
| string | customer_id | Customer ID |
| string | store       | Store ID    |

#### Faults:

| Fault Code | Fault Message                                     |
|------------|---------------------------------------------------|
| 101        | Requested store does not exist.                   |
| 102        | Requested product does not exist.                 |
| 103        | Requested customer does not exist.                |
| 105        | Provided data is invalid.                         |
| 106        | Error while saving tag. Details in error message. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$data = array('product_id' => 2, 'store' => 'default', 'customer_id' => 10002, 'tag' => "First 'Second tag' Third");
echo "Adding Tag... ";
$addResult = $proxy->call(
 $sessionId,
 "product_tag.add",
 array($data)
);
echo ((count($addResult) == 3) ? "Done!" : "Fail!");
echo "
";
print_r($addResult);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->catalogProductTagAdd($sessionId, array(
 'tag' => 'album',
 'product_id' => '3',
 'customer_id' => '1',
 'store' => '0'
));
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductTagAdd((object)array('sessionId' => $sessionId->result, 'data' => ((object)array(
 'tag' => 'album',
```

```
'product_id' => '3',
'customer_id' => '1',
'store' => '0'
)));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Tag Info

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[product\\_tag.add](#)

[product\\_tag.info](#)

[product\\_tag.list](#)

[product\\_tag.remove](#)

[product\\_tag.update](#)

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

## Catalog Inventory ▼

## Checkout ▼

## Create Your Own API

## Customer ▼

## Directory ▼

## Module: Tag API

Resource: catalog\_product\_tag

**Aliases:** product\_tag

### Method:

- catalog\_product\_tag.info (SOAP V1)
- catalogProductTagInfo (SOAP V2)

Allows you to retrieve information about the required product tag.

### Arguments:

| Type   | Name      | Description           |
|--------|-----------|-----------------------|
| string | sessionId | Session ID            |
| string | tagId     | Tag ID                |
| string | store     | Store view code or ID |

### Return:

| Type  | Name   | Description                          |
|-------|--------|--------------------------------------|
| array | result | Array of catalogProductTagInfoEntity |

The `catalogProductTagInfoEntity` content is as follows:

| Type   | Name   | Description |
|--------|--------|-------------|
| string | name   | Tag name    |
| string | status | Tag status  |

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

|                  |                 |                                                                                                 |
|------------------|-----------------|-------------------------------------------------------------------------------------------------|
| string           | base_popularity | Tag base popularity for a specific store                                                        |
| associativeArray | products        | Associative array of tagged products with related product ID as a key and popularity as a value |

#### Faults:

| Fault Code | Fault Message                   |
|------------|---------------------------------|
| 101        | Requested store does not exist. |
| 104        | Requested tag does not exist.   |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_tag.info', array('tagId' => '4',
'store' => '2'));
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductTagInfo($sessionId, '4', '2');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductTagInfo((object)array('sessionId' => $sessionId-
>result, 'tagId' => '4', 'store' => '2'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 'status' => string '1' (length=1)
 'name' => string 'digital' (length=7)
 'base_popularity' => int 0
 'products' =>
 array
 1 => string '1' (length=1)
 3 => string '1' (length=1)
 4 => string '1' (length=1)
```



## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Tag List

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[product\\_tag.add](#)

[product\\_tag.info](#)

[product\\_tag.list](#)

[product\\_tag.remove](#)

[product\\_tag.update](#)

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

## Catalog Inventory ▼

## Checkout ▼

## Create Your Own API

## Customer ▼

## Directory ▼

## Module: Tag API

Resource: catalog\_product\_tag

**Aliases:** product\_tag

### Method:

- catalog\_product\_tag.list (SOAP V1)
- catalogProductTagList (SOAP V2)

Allows you to retrieve the list of tags for a specific product.

### Arguments:

| Type   | Name      | Description           |
|--------|-----------|-----------------------|
| string | sessionId | Session ID            |
| string | productId | Product ID            |
| string | store     | Store view code or ID |

### Return:

| Type  | Name   | Description                          |
|-------|--------|--------------------------------------|
| array | result | Array of catalogProductTagListEntity |

The **catalogProductTagListEntity** content is as follows:

| Type   | Name   | Description |
|--------|--------|-------------|
| string | tag_id | Tag ID      |
| string | name   | Tag name    |

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

#### Faults:

| Fault Code | Fault Message                     |
|------------|-----------------------------------|
| 101        | Requested store does not exist.   |
| 102        | Requested product does not exist. |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_tag.list', array('productId' =>
'4', 'store' => '2'));
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductTagList($sessionId, '4', '2');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductTagList((object)array('sessionId' => $sessionId-
>result, 'productId' => '4', 'store' => '2'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 3 =>
 array
 'tag_id' => string '3' (length=1)
 'name' => string 'canon' (length=5)
 4 =>
 array
 'tag_id' => string '4' (length=1)
 'name' => string 'digital' (length=7)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Tag Remove

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[product\\_tag.add](#)

[product\\_tag.info](#)

[product\\_tag.list](#)

[product\\_tag.remove](#)

[product\\_tag.update](#)

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

## Catalog Inventory ^

## Checkout ^

## Create Your Own API

## Customer ^

## Directory ^

## Module: Tag API

Resource: catalog\_product\_tag

**Aliases:** product\_tag

### Method:

- catalog\_product\_tag.remove (SOAP V1)
- catalogProductTagRemove (SOAP V2)

Allows you to remove an existing product tag.

### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |
| string | tagId     | Tag ID      |

### Return:

| Type        | Description                            |
|-------------|----------------------------------------|
| boolean\int | True (1) if the product tag is removed |

### Faults:

| Fault Code | Fault Message                                       |
|------------|-----------------------------------------------------|
| 104        | Requested tag does not exist.                       |
| 107        | Error while removing tag. Details in error message. |

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_tag.remove', '3');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductTagRemove($sessionId, '3');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductTagRemove((object)array('sessionId' => $sessionId-
>result, 'tagId' => '3'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Tag Update

[Edit this page on GitHub](#) 

## Introduction

## Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[product\\_tag.add](#)

[product\\_tag.info](#)

[product\\_tag.list](#)

[product\\_tag.remove](#)

[product\\_tag.update](#)

[Catalog Product Tier Price](#) ▼

[Catalog Product Types](#) ▼

## Catalog Inventory ▼

## Checkout ▼

## Create Your Own API

## Customer ▼

## Directory ▼

## Module: Tag API

Resource: catalog\_product\_tag

**Aliases:** product\_tag

### Method:

- catalog\_product\_tag.update (SOAP V1)
- catalogProductTagUpdate (SOAP V2)

Allows you to update information about an existing product tag.

### Arguments:

| Type   | Name      | Description                                                         |
|--------|-----------|---------------------------------------------------------------------|
| string | sessionId | Session ID                                                          |
| string | tagId     | ID of the tag to be updated                                         |
| array  | data      | Array of catalogProductTagUpdateEntity                              |
| string | store     | Store view code or ID (optional; required for WS-I compliance mode) |

### Return:

| Type    | Description                        |
|---------|------------------------------------|
| boolean | True if the product tag is updated |

The **catalogProductTagUpdateEntity** content is as follows:

| Type   | Name | Description |
|--------|------|-------------|
| string | name | Tag name    |

Sales ▾

Enterprise Customer Balance ▾

Enterprise Gift Card ▾

Enterprise Gift Message ▾

Miscellaneous ▾

WS-I Compliance

|                |                 |                                                                                    |
|----------------|-----------------|------------------------------------------------------------------------------------|
| string         | status          | Tag status. Can have the following values: -1 - Disabled, 0 - Pending, 1- Approved |
| string         | base_popularity | Tag base popularity                                                                |
| <b>Faults:</b> |                 |                                                                                    |
|                | Fault Code      | Fault Message                                                                      |
|                | 101             | Requested store does not exist.                                                    |
|                | 104             | Requested tag does not exist.                                                      |
|                | 105             | Provided data is invalid.                                                          |
|                | 106             | Error while saving tag. Details in error message.                                  |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_tag.update', array('tagId' => '4',
'data' => array('name' => 'digital_1')));
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->catalogProductTagUpdate($sessionId, '1', array(
'name' => 'tag',
'status' => '1'
));
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductTagUpdate((object)array('sessionId' => $sessionId-
>result, 'tagId' => '1', 'store' => '0', 'data' => ((object)array(
'name' => 'tag',
'status' => '1',
'base_popularity' => null
))));
```

var\_dump(\$result->result);

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product Tier Price

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▾

[Catalog Category Attributes](#) ▾

[Catalog Product](#) ▾

[Catalog Product Attribute](#) ▾

[Catalog Product Attribute](#)

[Media](#) ▾

[Catalog Product Attribute Set](#) ▾

[Catalog Product Custom](#)

[Option](#) ▾

[Catalog Product Custom Option](#)

[Value](#) ▾

[Catalog Product Downloadable](#)

[Link](#) ▾

[Catalog Product Link](#) ▾

[Catalog Product Tag](#) ▾

[Catalog Product Tier Price](#) ^

[catalog\\_product\\_attribute\\_tier\\_...](#)

[catalog\\_product\\_attribute\\_tier\\_...](#)

[Catalog Product Types](#) ▾

### Catalog Inventory ▾

### Checkout ▾

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Product Tier Price

Allows you to retrieve and update product tier prices.

**Resource Name:** catalog\_product\_attribute\_tier\_price

#### Aliases:

- product\_attribute\_tier\_price
- product\_tier\_price

#### Methods:

- [catalog\\_product\\_attribute\\_tier\\_price.info](#) - Retrieve information about product tier prices
- [catalog\\_product\\_attribute\\_tier\\_price.update](#) - Update the product tier prices

### Faults

| Fault Code | Fault Message                                      |
|------------|----------------------------------------------------|
| 100        | Product not exists.                                |
| 101        | Invalid data given. Details in error message.      |
| 102        | Tier prices not updated. Details in error message. |

### Examples

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
```

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

---

```
// Get tier prices
$tierPrices = $proxy->call($sessionId, 'product_tier_price.info', 'Sku');

var_dump($tierPrices);

// Add new
$tierPrices[] = array(
 'website' => 'all',
 'customer_group_id' => 'all',
 'qty' => 68,
 'price' => 18.20
);

// Update tier prices
$proxy->call($sessionId, 'product_tier_price.update', array('Sku', $tierPrices));

// Compare values
var_dump($proxy->call($sessionId, 'product_tier_price.info', 'Sku'));

var_dump($tierPrices);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Attribute Tier Price Info

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▼

[Catalog Category Attributes](#) ▼

[Catalog Product](#) ▼

[Catalog Product Attribute](#) ▼

[Catalog Product Attribute](#)

[Media](#) ▼

[Catalog Product Attribute Set](#) ▼

[Catalog Product Custom](#)

[Option](#) ▼

[Catalog Product Custom Option](#)

[Value](#) ▼

[Catalog Product Downloadable](#)

[Link](#) ▼

[Catalog Product Link](#) ▼

[Catalog Product Tag](#) ▼

[Catalog Product Tier Price](#) ▼

[catalog\\_product\\_attribute\\_tier...](#)

[catalog\\_product\\_attribute\\_tier...](#)

[Catalog Product Types](#) ▼

[Catalog Inventory](#) ▼

[Checkout](#) ▼

[Create Your Own API](#)

[Customer](#) ▼

[Directory](#) ▼

[Sales](#) ▼

[Enterprise Customer Balance](#) ▼

## Module: Mage\_Catalog

Resource: catalog\_product\_attribute\_tier\_price

### Aliases:

- product\_attribute\_tier\_price
- product\_tier\_price

### Method:

- catalog\_product\_attribute\_tier\_price.info (SOAP V1)
- catalogProductAttributeTierPriceInfo (SOAP V2)

Allows you to retrieve information about product tier prices.

### Aliases:

- product\_attribute\_tier\_price.info
- product\_tier\_price.info

### Arguments:

| Type   | Name              | Description                                                                |
|--------|-------------------|----------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                 |
| string | product\productId | Product ID or SKU                                                          |
| string | identifierType    | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

| Type  | Name   | Description                            |
|-------|--------|----------------------------------------|
| array | result | Array of catalogProductTierPriceEntity |

[Enterprise Gift Card](#) ✓

[Enterprise Gift Message](#) ✓

[Miscellaneous](#) ✓

[WS-I Compliance](#)

The **catalogProductTierPriceEntity** content is as follows:

| Type   | Name              | Description                                          |
|--------|-------------------|------------------------------------------------------|
| string | customer_group_id | Customer group ID                                    |
| string | website           | Website                                              |
| int    | qty               | Quantity of items to which the price will be applied |
| double | price             | Price that each item will cost                       |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_attribute_tier_price.info',
'productId');
var_dump($result);
```

### Request Example SOAP V2

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$productId = 1;

$result = $client->catalogProductAttributeTierPriceInfo(
 $session,
 $productId
);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogProductAttributeTierPriceInfo((object)array('sessionId' =>
$sessionId->result, 'productId' => '1'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
 array
```

```
'customer_group_id' => string '1' (length=1)
'website' => string 'all' (length=3)
'qty' => string '2.0000' (length=6)
'price' => string '129.9900' (length=8)
```

## Response Example SOAP V2

```
array
0 =>
object(stdClass)[2]
 public 'customer_group_id' => string '0' (length=1)
 public 'website' => string 'all' (length=3)
 public 'qty' => int 5
 public 'price' => float 99
1 =>
object(stdClass)[3]
 public 'customer_group_id' => string '0' (length=1)
 public 'website' => string 'all' (length=3)
 public 'qty' => int 10
 public 'price' => float 98
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Attribute Tier Price Update

[Edit this page on GitHub](#) 

## Introduction

### Catalog ^

[Catalog Category](#) ▾

[Catalog Category Attributes](#) ▾

[Catalog Product](#) ▾

[Catalog Product Attribute](#) ▾

[Catalog Product Attribute](#)

[Media](#) ▾

[Catalog Product Attribute Set](#) ▾

[Catalog Product Custom](#)

[Option](#) ▾

[Catalog Product Custom Option](#)

[Value](#) ▾

[Catalog Product Downloadable](#)

[Link](#) ▾

[Catalog Product Link](#) ▾

[Catalog Product Tag](#) ▾

[Catalog Product Tier Price](#) ▾

[catalog\\_product\\_attribute\\_tier...](#)

[catalog\\_product\\_attribute\\_tier...](#)

[Catalog Product Types](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

## Module: Mage\_Catalog

Resource: catalog\_product\_attribute\_tier\_price

### Aliases:

- product\_attribute\_tier\_price
- product\_tier\_price

### Method:

- catalog\_product\_attribute\_tier\_price.update (SOAP V1)
- catalogProductAttributeTierPriceUpdate (SOAP V2)

Allows you to update the product tier prices.

### Aliases:

- product\_attribute\_tier\_price.update
- product\_tier\_price.update

### Arguments:

| Type   | Name              | Description                                                                |
|--------|-------------------|----------------------------------------------------------------------------|
| string | sessionId         | Session ID                                                                 |
| string | product\productId | Product ID or SKU                                                          |
| array  | tierPrices        | Array of catalogProductTierPriceEntity                                     |
| string | identifierType    | Defines whether the product ID or SKU is passed in the 'product' parameter |

### Returns:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

[Enterprise Gift Card](#) ✓

[Enterprise Gift Message](#) ✓

[Miscellaneous](#) ✓

[WS-I Compliance](#)

|             |        |                                               |
|-------------|--------|-----------------------------------------------|
| boolean\int | result | True (1) if the product tier price is updated |
|-------------|--------|-----------------------------------------------|

The **catalogProductTierPriceEntity** content is as follows:

| Type   | Name              | Description                                          |
|--------|-------------------|------------------------------------------------------|
| string | customer_group_id | Customer group ID                                    |
| string | website           | Website                                              |
| int    | qty               | Quantity of items to which the price will be applied |
| double | price             | Price that each item will cost                       |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$productId = 1;
$tierPrices = array(
 array('customer_group_id' => '0', 'website' => '0', 'qty' => '50', 'price' =>
'9.90')
);

$result = $proxy->call(
 $sessionId,
 'product_attribute_tier_price.update',
 array(
 $productId,
 $tierPrices
)
);

var_dump($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$productId = 1;
$tierPrices = array(
 array('customer_group_id' => '0', 'website' => '0', 'qty' => '50', 'price' =>
'9.90')
);

$result = $proxy->catalogProductAttributeTierPriceUpdate(
 $sessionId,
 $productId,
 $tierPrices
);

var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$productId = 1;
$tierPrices = array(
 array('customer_group_id' => '0', 'website' => '0', 'qty' => '50', 'price' =>
'9.90')
);

$result = $proxy->catalogProductAttributeTierPriceUpdate((object)array(
'sessionId' => $sessionId->result,
'productId' => $productId,
'tierPrices' => $tierPrices
));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Product Types

[Edit this page on GitHub](#) 

## Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

`catalog_product_type.list`

## Catalog Inventory

### Checkout

## Create Your Own API

### Customer

### Directory

### Sales

### Enterprise Customer Balance

### Enterprise Gift Card

## Module: Mage\_Catalog

The Mage\_Catalog module allows you to manage categories and products.

### Product types

Allows you to retrieve product types.

**Resource Name:** catalog\_product\_type

#### Aliases:

- product\_type

#### Methods:

- [catalog\\_product\\_type.list](#) - Retrieve the list of product types

### Examples

#### Example 1. Retrieving the product types

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$types = $proxy->call($sessionId, 'product_type.list');

var_dump($types);
```

Create the Magento file system owner

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

---

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Type List

[Edit this page on GitHub](#) 

### Introduction

### Catalog

[Catalog Category](#) 

[Catalog Category Attributes](#) 

[Catalog Product](#) 

[Catalog Product Attribute](#) 

[Catalog Product Attribute](#)

[Media](#) 

[Catalog Product Attribute Set](#) 

[Catalog Product Custom](#)

[Option](#) 

[Catalog Product Custom Option](#)

[Value](#) 

[Catalog Product Downloadable](#)

[Link](#) 

[Catalog Product Link](#) 

[Catalog Product Tag](#) 

[Catalog Product Tier Price](#) 

[Catalog Product Types](#) 

[catalog\\_product\\_type.list](#)

[Catalog Inventory](#) 

[Checkout](#) 

[Create Your Own API](#)

[Customer](#) 

[Directory](#) 

[Sales](#) 

[Enterprise Customer Balance](#) 

[Enterprise Gift Card](#) 

## Module: Mage\_Catalog

Resource: catalog\_product\_type

### Aliases:

- product\_type

### Method:

- catalog\_product\_type.list (SOAP V1)
- catalogProductTypeList (SOAP V2)

Allows you to retrieve the list of product types.

### Aliases:

- product\_type.list

### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

### Returns:

Type Name Description

array result Array of catalogProductTypeEntity

The `catalogProductTypeEntity` content is as follows:

Type Name Description

string type Product type

**Enterprise Gift Message** ▾

|        |       |                                  |
|--------|-------|----------------------------------|
| string | label | Product label in the Admin Panel |
|--------|-------|----------------------------------|

**Miscellaneous** ▾**WS-I Compliance**

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'catalog_product_type.list');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogProductTypeList($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogProductTypeList((object)array('sessionId' => $sessionId->result));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'type' => string 'simple' (length=6)
 'label' => string 'Simple Product' (length=14)
1 =>
array
 'type' => string 'grouped' (length=7)
 'label' => string 'Grouped Product' (length=15)
2 =>
array
 'type' => string 'configurable' (length=12)
 'label' => string 'Configurable Product' (length=20)
3 =>
array
 'type' => string 'virtual' (length=7)
 'label' => string 'Virtual Product' (length=15)
4 =>
array
```

```
'type' => string 'bundle' (length=6)
'label' => string 'Bundle Product' (length=14)
5 =>
array
'type' => string 'downloadable' (length=12)
'label' => string 'Downloadable Product' (length=20)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Catalog Inventory

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

### Catalog Inventory ▾

[cataloginventory\\_stock\\_item.list](#)  
[cataloginventory\\_stock\\_item.update](#)

## Checkout ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Module: Mage\_CatalogInventory

The Mage\_CatalogInventory module allows you to retrieve and update stock attributes, such as status and quantity.

### Inventory

Allows you to retrieve and update the stock data.

**Resource Name:** cataloginventory\_stock\_item

#### Aliases:

- product\_stock

#### Methods:

- [cataloginventory\\_stock\\_item.list](#) - Retrieve the list of stock data by product IDs
- [cataloginventory\\_stock\\_item.update](#) - Update the stock data for a list of products

### Faults

| Fault Code | Fault Message                                            |
|------------|----------------------------------------------------------|
| 101        | Product not exists.                                      |
| 102        | Product inventory not updated. Details in error message. |

### Examples

Change manage\_stock setting to 'off' in the inventory area.

```
$attributeSets = $client->call($session, 'product_stock.update',
array('SKU',array('manage_stock'=>'0','use_config_manage_stock'=>'0')));
```

The use\_config\_manage\_stock unchecks the 'Use Config Settings' box which allows you to make changes to

this product and not to use the global settings that are set by default.

### Example 1. Working with stock update

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

// Get stock info
var_dump($proxy->call($sessionId, 'product_stock.list', 'Sku'));

// Update stock info
$proxy->call($sessionId, 'product_stock.update', array('Sku', array('qty'=>50,
'is_in_stock'=>1)));

var_dump($proxy->call($sessionId, 'product_stock.list', 'Sku'));
```

### Example 2. Listing multiple SKUs' data

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');
$sid = $client->login('apiUser', 'apiKey');

print_r(
 $client->call(
 $sid,
 'product_stock.list',
 array(
 array(// Notice the nested array
 'sku1',
 'sku2',
 'sku3',
 ...
 'skuN'
)
)
)
);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Inventory Item List

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▲

[cataloginventory\\_stock\\_item.list](#)

[cataloginventory\\_stock\\_item.update](#)

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Module: Mage\_CatalogInventory

Resource: cataloginventory\_stock\_item

**Aliases:**

- product\_stock

**Method:**

- cataloginventory\_stock\_item.list (SOAP V1)
- catalogInventoryStockItemList (SOAP V2)

Allows you to retrieve the list of stock data by product IDs.

**Aliases:**

- product\_stock.list

**Arguments:**

| Type          | Name                                | Description                 |
|---------------|-------------------------------------|-----------------------------|
| string        | sessionId                           | Session ID                  |
| ArrayOfString | products/productIds (for WS-I mode) | List of product IDs or SKUs |

**Returns:**

| Type  | Name   | Description                              |
|-------|--------|------------------------------------------|
| array | result | Array of catalogInventoryStockItemEntity |

The **catalogInventoryStockItemEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |             |                                         |
|--------|-------------|-----------------------------------------|
| string | product_id  | Product ID                              |
| string | sku         | Product SKU                             |
| string | qty         | Product quantity                        |
| string | is_in_stock | Defines whether the product is in stock |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'cataloginventory_stock_item.list', '1');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->catalogInventoryStockItemList($sessionId, array('1', '2'));
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->catalogInventoryStockItemList((object)array('sessionId' => $sessionId->result, 'productIds' => array(1,2)));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 0 =>
 array
 'product_id' => string '1' (length=1)
 'sku' => string 'n2610' (length=5)
 'qty' => string '98.0000' (length=7)
 'is_in_stock' => string '1' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Inventory Item Update

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▲

[cataloginventory\\_stock\\_item.list](#)

[cataloginventory\\_stock\\_item.update](#)

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Module: Mage\_CatalogInventory

Resource: cataloginventory\_stock\_item

**Aliases:**

- product\_stock

**Method:**

- cataloginventory\_stock\_item.update (SOAP V1)
- catalogInventoryStockItemUpdate (SOAP V2)

Allows you to update the required product stock data.

**Aliases:**

- product\_stock.update

**Arguments:**

| Type   | Name              | Description                                    |
|--------|-------------------|------------------------------------------------|
| string | sessionId         | Session ID                                     |
| string | product\productId | Product ID or SKU                              |
| array  | data              | Array of catalogInventoryStockItemUpdateEntity |

**Returns:**

| Type | Name   | Description                   |
|------|--------|-------------------------------|
| int  | result | Result of stock item updating |

The **catalogInventoryStockItemUpdateEntity** content is as follows:

| Type   | Name                        | Description                                                                    |
|--------|-----------------------------|--------------------------------------------------------------------------------|
| string | qty                         | Quantity of items to be updated                                                |
| int    | is_in_stock                 | Defines whether the item is in stock                                           |
| int    | manage_stock                | Manage stock flag                                                              |
| int    | use_config_manage_stock     | Use config manage stock                                                        |
| int    | min_qty                     | Minimum quantity for items to be in stock                                      |
| int    | use_config_min_qty          | Use config settings flag (value defined in the Inventory System Configuration) |
| int    | min_sale_qty                | Minimum quantity allowed in the shopping cart                                  |
| int    | use_config_min_sale_qty     | Use config settings flag                                                       |
| int    | max_sale_qty                | Maximum quantity allowed in the shopping cart                                  |
| int    | use_config_max_sale_qty     | Use config settings flag                                                       |
| int    | is_qty_decimal              | Defines whether the quantity is decimal                                        |
| int    | backorders                  | Backorders status                                                              |
| int    | use_config_backorders       | Use config settings flag (for backorders)                                      |
| int    | notify_stock_qty            | Stock quantity below which a notification will appear                          |
| int    | use_config_notify_stock_qty | Use config settings flag (for stock quantity)                                  |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$productID = '1';
$stockItemData = array(
 'qty' => '100',
 'is_in_stock' => 1,
 'manage_stock' => 1,
 'use_config_manage_stock' => 0,
 'min_qty' => 2,
 'use_config_min_qty' => 0,
 'min_sale_qty' => 1,
 'use_config_min_sale_qty' => 0,
 'max_sale_qty' => 10,
 'use_config_max_sale_qty' => 0,
 'is_qty_decimal' => 0,
 'backorders' => 1,
 'use_config_backorders' => 0,
 'notify_stock_qty' => 10,
 'use_config_notify_stock_qty' => 0
);

$result = $client->call(
```

```
 $session,
 'product_stock.update',
 array(
 $productId,
 $stockItemData
)
);
}
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->catalogInventoryStockItemUpdate($sessionId, 1, array(
 'qty' => '49',
 'is_in_stock' => 1
));

var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->catalogInventoryStockItemUpdate((object)array('sessionId' =>
$sessionId->result, 'productId' => '1', 'data' => array(
 'qty' => '49',
 'is_in_stock' => 1
)));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Checkout

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▲

**Cart** ▾

**Cart Coupon** ▾

**Cart Customer** ▾

**Cart Payment** ▾

**Cart Product** ▾

**Cart Shipping** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

## Module: Mage\_Checkout

The Mage\_Checkout module allows you to manage shopping carts and the checkout process. This module allows you to create an order once filling the shopping cart is complete.

### Cart Coupon

Allows you to add and remove coupon codes for a shopping cart.

**Resource Name:** cart\_coupon

**Methods:**

- [cart\\_coupon.add](#) - Add a coupon code to a quote
- [cart\\_coupon.remove](#) - Remove a coupon code from a quote

### Cart Customer

Allows you to add customer information and addresses into a shopping cart.

**Resource Name:** cart\_customer

**Methods:**

- [cart\\_customer.set](#) - Add customer information into a shopping cart
- [cart\\_customer.addresses](#) - Set the customer addresses (shipping and billing) into a shopping cart

### Cart Payment

Allows you to retrieve and set payment methods for a shopping cart.

**Resource Name:** cart\_payment

**Methods:**

- [cart\\_payment.method](#) - Set a payment method for a shopping cart

- [cart\\_payment.list](#) - Get the list of available payment methods for a shopping cart

## Cart Product

Allows you to manage products in a shopping cart.

**Resource Name:** cart\_product

**Methods:**

- [cart\\_product.add](#) - Add one or more products to a shopping cart
- [cart\\_product.update](#) - Update one or more products in a shopping cart
- [cart\\_product.remove](#) - Remove one or more products from a shopping cart
- [cart\\_product.list](#) - Get a list of products in a shopping cart
- [cart\\_product.moveToCustomerQuote](#) - Move one or more products from the quote to the customer shopping cart

## Cart Shipping

Allows you to retrieve and set shipping methods for a shopping cart.

**Resource Name:** cart\_shipping

**Methods:**

- [cart\\_shipping.method](#) - Set a shipping method for a shopping cart
- [cart\\_shipping.list](#) - Retrieve the list of available shipping methods for a shopping cart

## Shopping Cart

Allows you to manage shopping carts.

**Resource Name:** cart

**Methods:**

- [cart.create](#) - Create a blank shopping cart
- [cart.order](#) - Create an order from a shopping cart
- [cart.info](#) - Get full information about the current shopping cart
- [cart.totals](#) - Get all available prices for items in shopping cart, using additional parameters
- [cart.license](#) - Get website license agreement

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Cart

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

### Cart ▾

cart.create  
cart.info  
cart.license  
cart.order  
cart.totals

### Cart Coupon ▾

### Cart Customer ▾

### Cart Payment ▾

### Cart Product ▾

### Cart Shipping ▾

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Checkout

The Mage\_Checkout module allows you to manage shopping carts and the checkout process. This module allows you to create an order once filling the shopping cart is complete.

### Shopping Cart

Allows you to manage shopping carts.

**Resource Name:** cart

#### Methods:

- [cart.create](#) - Create a blank shopping cart
- [cart.order](#) - Create an order from a shopping cart
- [cart.info](#) - Get full information about the current shopping cart
- [cart.totals](#) - Get all available prices for items in shopping cart, using additional parameters
- [cart.licenseAgreement](#) - Get website license agreement

### Faults

| Fault Code | Fault Message                                                               |
|------------|-----------------------------------------------------------------------------|
| 1001       | Can not make operation because store is not exists                          |
| 1002       | Can not make operation because quote is not exists                          |
| 1003       | Can not create a quote.                                                     |
| 1004       | Can not create a quote because quote with such identifier is already exists |
| 1005       | You did not set all required agreements                                     |
| 1006       | The checkout type is not valid. Select single checkout type.                |

|      |                                     |
|------|-------------------------------------|
| 1007 | Checkout is not available for guest |
| 1008 | Can not create an order.            |

## Example

The following example illustrates the work with shopping cart (creation of a shopping cart, setting customer and customer addresses, adding products to the shopping cart, updating products in the shopping cart, removing products from the shopping cart, getting the list of products/shipping methods/payment methods, setting payment/shipping methods, adding/removing coupon, getting total prices/full information about shopping cart/list of licenses, and creating an order).

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

// Create a quote, get quote identifier
$shoppingCartId = $proxy->call($sessionId, 'cart.create', array('magento_store'));

// Set customer, for example guest
$customerAsGuest = array(
 "firstname" => "testFirstname",
 "lastname" => "testLastName",
 "email" => "testEmail",
 "website_id" => "0",
 "store_id" => "0",
 "mode" => "guest"
);
$resultCustomerSet = $proxy->call($sessionId, 'cart_customer.set', array(
$shoppingCartId, $customerAsGuest));

// Set customer addresses, for example guest's addresses
$arrAddresses = array(
 array(
 "mode" => "shipping",
 "firstname" => "testFirstname",
 "lastname" => "testLastname",
 "company" => "testCompany",
 "street" => "testStreet",
 "city" => "testCity",
 "region" => "testRegion",
 "postcode" => "testPostcode",
 "country_id" => "id",
 "telephone" => "0123456789",
 "fax" => "0123456789",
 "is_default_shipping" => 0,
 "is_default_billing" => 0
),
 array(
 "mode" => "billing",
 "firstname" => "testFirstname",
 "lastname" => "testLastname",
 "company" => "testCompany",
 "street" => "testStreet",
 "city" => "testCity",
 "region" => "testRegion",
 "postcode" => "testPostcode",
 "country_id" => "id",
 "telephone" => "0123456789",
 "fax" => "0123456789",
 "is_default_shipping" => 0,
 "is_default_billing" => 0
)
)
```

```

);

$resultCustomerAddresses = $proxy->call($sessionId, "cart_customer.addresses",
array($shoppingCartId, $arrAddresses));

// add products into shopping cart
$arrProducts = array(
 array(
 "product_id" => "1",
 "qty" => 2
),
 array(
 "sku" => "testSKU",
 "quantity" => 4
)
);
$resultCartProductAdd = $proxy->call($sessionId, "cart_product.add",
array($shoppingCartId, $arrProducts));

// update product in shopping cart
$arrProducts = array(
 array(
 "product_id" => "1",
 "qty" => 5
),
);
$resultCartProductUpdate = $proxy->call($sessionId, "cart_product.update",
array($shoppingCartId, $arrProducts));

// remove products from shopping cart, for example by SKU
$arrProducts = array(
 array(
 "sku" => "testSKU"
),
);
$resultCartProductRemove = $proxy->call($sessionId, "cart_product.remove",
array($shoppingCartId, $arrProducts));

// get list of products
$shoppingCartProducts = $proxy->call($sessionId, "cart_product.list",
array($shoppingCartId));
print_r($shoppingCartProducts);

// get list of shipping methods
$resultShippingMethods = $proxy->call($sessionId, "cart_shipping.list",
array($shoppingCartId));
print_r($resultShippingMethods);

// set shipping method
$randShippingMethodIndex = rand(1, count($resultShippingMethods));
$shippingMethod = $resultShippingMethods[$randShippingMethodIndex] ["code"];

$resultShippingMethod = $proxy->call($sessionId, "cart_shipping.method",
array($shoppingCartId, $shippingMethod));

// get list of payment methods
$resultPaymentMethods = $proxy->call($sessionId, "cart_payment.list",
array($shoppingCartId));
print_r($resultPaymentMethods);

// set payment method
$paymentMethod = array(
 "method" => "checkmo"
);

```

```
$resultPaymentMethod = $proxy->call($sessionId, "cart_payment.method",
array($shoppingCartId, $paymentMethod));

// add coupon
$couponCode = "aCouponCode";
$resultCartCouponRemove = $proxy->call($sessionId, "cart_coupon.add",
array($shoppingCartId, $couponCode));

// remove coupon
$resultCartCouponRemove = $proxy->call($sessionId, "cart_coupon.remove",
array($shoppingCartId));

// get total prices
$shoppingCartTotals = $proxy->call($sessionId, "cart.totals", array($shoppingCartId));
print_r($shoppingCartTotals);

// get full information about shopping cart
$shoppingCartInfo = $proxy->call($sessionId, "cart.info", array($shoppingCartId));
print_r($shoppingCartInfo);

// get list of licenses
$shoppingCartLicenses = $proxy->call($sessionId, "cart.licenseAgreement",
array($shoppingCartId));
print_r($shoppingCartLicences);

// check if license is existed
$licenseForOrderCreation = null;
if (count($shoppingCartLicenses)) {
 $licenseForOrderCreation = array();
 foreach ($shoppingCartLicenses as $license) {
 $licenseForOrderCreation[] = $license['agreement_id'];
 }
}

// create order
$resultOrderCreation = $proxy->call($sessionId, "cart.order", array($shoppingCartId,
null, $licenseForOrderCreation));
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Cart Create

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

### Cart ▾

- cart.create
- cart.info
- cart.license
- cart.order
- cart.totals

## Cart Coupon ▾

## Cart Customer ▾

## Cart Payment ▾

## Cart Product ▾

## Cart Shipping ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Module: Mage\_Checkout

Resource: cart

### Method:

- cart.create (SOAP V1)
- shoppingCartCreate (SOAP V2)

Allows you to create an empty shopping cart.

### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | storeId   | Store view ID or code (optional) |

### Returns:

| Type | Description                           |
|------|---------------------------------------|
| int  | ID of the created empty shopping cart |

### Faults:

No Faults

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
```

```
$shoppingCartIncrementId = $proxy->call($sessionId, 'cart.create', array(
'magento_store'));
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if
necesary

$result = $proxy->shoppingCartCreate($sessionId, '3');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartCreate((object)array('sessionId' => $sessionId->result,
'store' => '3'));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Cart Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Cart ▾

cart.create  
cart.info  
cart.license  
cart.order  
cart.totals

### Cart Coupon ▾

### Cart Customer ▾

### Cart Payment ▾

### Cart Product ▾

### Cart Shipping ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Mage\_Checkout

### Module: Shopping Cart API

Resource: cart

#### Method:

- cart.info (SOAP V1)
- shoppingCartInfo (SOAP V2)

Allows you to retrieve full information about the shopping cart (quote).

#### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| int    | quoteId   | Shopping cart ID (quote ID)      |
| string | store     | Store view ID or code (optional) |

#### Return:

| Type  | Name   | Description                     |
|-------|--------|---------------------------------|
| array | result | Array of shoppingCartInfoEntity |

The **shoppingCartInfoEntity** content is as follows:

| Type   | Name     | Description |
|--------|----------|-------------|
| string | store_id | Store ID    |

|        |                       |                                              |
|--------|-----------------------|----------------------------------------------|
| string | created_at            | Date of creation                             |
| string | updated_at            | Date of updating                             |
| string | converted_at          | Date of conversion                           |
| int    | quote_id              | Quote ID                                     |
| int    | is_active             | Active flag                                  |
| int    | is_virtual            | Defines whether the product is a virtual one |
| int    | is_multi_shipping     | Defines whether multi shipping is available  |
| double | items_count           | Items quantity                               |
| double | items_qty             | Total items quantity                         |
| string | orig_order_id         | Original order ID                            |
| string | store_to_base_rate    | Store to base rate                           |
| string | store_to_quote_rate   | Store to quote rate                          |
| string | base_currency_code    | Base currency code                           |
| string | store_currency_code   | Store currency code                          |
| string | quote_currency_code   | Quote currency code                          |
| string | grand_total           | Grand total                                  |
| string | base_grand_total      | Base grand total                             |
| string | checkout_method       | Checkout method                              |
| string | customer_id           | Customer ID                                  |
| string | customer_tax_class_id | Customer tax class ID                        |
| int    | customer_group_id     | Customer group ID                            |
| string | customer_email        | Customer email address                       |
| string | customer_prefix       | Customer prefix                              |
| string | customer_firstname    | Customer first name                          |
| string | customer_middlename   | Customer middle name                         |
| string | customer_lastname     | Customer last name                           |
| string | customer_suffix       | Customer suffix                              |
| string | customer_note         | Customer note                                |
| string | customer_note_notify  | Customer notification flag                   |
| string | customer_is_guest     | Defines whether the customer is a guest      |
| string | applied_rule_ids      | Applied rule IDs                             |
| string | reserved_order_id     | Reserved order ID                            |

|        |                                   |                                             |
|--------|-----------------------------------|---------------------------------------------|
| string | password_hash                     | Password hash                               |
| string | coupon_code                       | Coupon code                                 |
| string | global_currency_code              | Global currency code                        |
| double | base_to_global_rate               | Base to global rate                         |
| double | base_to_quote_rate                | Base to quote rate                          |
| string | customer_taxvat                   | Customer taxvat value                       |
| string | customer_gender                   | Customer gender                             |
| double | subtotal                          | Subtotal                                    |
| double | base_subtotal                     | Base subtotal                               |
| double | subtotal_with_discount            | Subtotal with discount                      |
| double | base_subtotal_with_discount       | Base subtotal with discount                 |
| string | ext_shipping_info                 |                                             |
| string | gift_message_id                   | Gift message ID                             |
| string | gift_message                      | Gift message                                |
| double | customer_balance_amount_used      | Used customer balance amount                |
| double | base_customer_balance_amount_used | Used base customer balance amount           |
| string | use_customer_balance              | Defines whether to use the customer balance |
| string | gift_cards_amount                 | Gift cards amount                           |
| string | base_gift_cards_amount            | Base gift cards amount                      |
| string | gift_cards_amount_used            | Used gift cards amount                      |
| string | use_reward_points                 | Defines whether to use reward points        |
| string | reward_points_balance             | Reward points balance                       |
| string | base_reward_currency_amount       | Base reward currency amount                 |
| string | reward_currency_amount            | Reward currency amount                      |
| array  | shipping_address                  | Array of shoppingCartAddressEntity          |
| array  | billing_address                   | Array of shoppingCartAddressEntity          |
| array  | items                             | Array of shoppingCartItemEntity             |
| array  | payment                           | Array of shoppingCartPaymentEntity          |

The **shoppingCartAddressEntity** content is as follows:

| Type   | Name       | Description              |
|--------|------------|--------------------------|
| string | address_id | Shopping cart address ID |
| string | created_at | Date of creation         |
| string | updated_at | Date of updating         |

|        |                      |                                                            |
|--------|----------------------|------------------------------------------------------------|
| string | customer_id          | Customer ID                                                |
| int    | save_in_address_book | Defines whether to save the address in the address book    |
| string | customer_address_id  | Customer address ID                                        |
| string | address_type         | Address type                                               |
| string | email                | Email address                                              |
| string | prefix               | Customer prefix                                            |
| string | firstname            | Customer first name                                        |
| string | middlename           | Customer middle name                                       |
| string | lastname             | Customer last name                                         |
| string | suffix               | Customer suffix                                            |
| string | company              | Company name                                               |
| string | street               | Street                                                     |
| string | city                 | City                                                       |
| string | region               | Region                                                     |
| string | region_id            | Region ID                                                  |
| string | postcode             | Postcode                                                   |
| string | country_id           | Country ID                                                 |
| string | telephone            | Telephone number                                           |
| string | fax                  | Fax                                                        |
| int    | same_as_billing      | Defines whether the address is the same as the billing one |
| int    | free_shipping        | Defines whether free shipping is used                      |
| string | shipping_method      | Shipping method                                            |
| string | shipping_description | Shipping description                                       |
| double | weight               | Weight                                                     |

The **shoppingCartItemEntity** content is as follows:

| Type   | Name           | Description      |
|--------|----------------|------------------|
| string | item_id        | Cart item ID     |
| string | created_at     | Date of creation |
| string | updated_at     | Date of updating |
| string | product_id     | Product ID       |
| string | store_id       | Store ID         |
| string | parent_item_id | Parent item ID   |

|        |                          |                                              |
|--------|--------------------------|----------------------------------------------|
| int    | is_virtual               | Defines whether the product is a virtual one |
| string | sku                      | Product SKU                                  |
| string | name                     | Product name                                 |
| string | description              | Description                                  |
| string | applied_rule_ids         | Applied rule IDs                             |
| string | additional_data          | Additional data                              |
| string | free_shipping            | Free shipping                                |
| string | is_qty_decimal           | Defines whether the quantity is decimal      |
| string | no_discount              | Defines whether no discount is applied       |
| double | weight                   | Weight                                       |
| double | qty                      | Quantity                                     |
| double | price                    | Price                                        |
| double | base_price               | Base price                                   |
| double | custom_price             | Custom price                                 |
| double | discount_percent         | Discount percent                             |
| double | discount_amount          | Discount amount                              |
| double | base_discount_amount     | Base discount amount                         |
| double | tax_percent              | Tax percent                                  |
| double | tax_amount               | Tax amount                                   |
| double | base_tax_amount          | Base tax amount                              |
| double | row_total                | Row total                                    |
| double | base_row_total           | Base row total                               |
| double | row_total_with_discount  | Row total with discount                      |
| double | row_weight               | Row weight                                   |
| string | product_type             | Product type                                 |
| double | base_tax_before_discount | Base tax before discount                     |
| double | tax_before_discount      | Tax before discount                          |
| double | original_custom_price    | Original custom price                        |
| double | base_cost                | Base cost                                    |
| double | price_incl_tax           | Price including tax                          |
| double | base_price_incl_tax      | Base price including tax                     |
| double | row_total_incl_tax       | Row total including tax                      |
| double | base_row_total_incl_tax  | Base row total including tax                 |

|        |                                   |                                                       |
|--------|-----------------------------------|-------------------------------------------------------|
| string | gift_message_id                   | Gift message ID                                       |
| string | gift_message                      | Gift message                                          |
| string | gift_message_available            | Defines whether the gift message is available         |
| double | weeee_tax_applied                 | Applied fix product tax                               |
| double | weeee_tax_applied_amount          | Applied fix product tax amount                        |
| double | weeee_tax_applied_row_amount      | Applied fix product tax row amount                    |
| double | base_weeee_tax_applied_amount     | Applied fix product tax amount (in base currency)     |
| double | base_weeee_tax_applied_row_amount | Applied fix product tax row amount (in base currency) |
| double | weeee_tax_disposition             | Fixed product tax disposition                         |
| double | weeee_tax_row_disposition         | Fixed product tax row disposition                     |
| double | base_weeee_tax_disposition        | Fixed product tax disposition (in base currency)      |
| double | base_weeee_tax_row_disposition    | Fixed product tax row disposition (in base currency)  |
| string | tax_class_id                      | Tax class ID                                          |

The **shoppingCartPaymentEntity** content is as follows:

| Type   | Name              | Description                            |
|--------|-------------------|----------------------------------------|
| string | payment_id        | Payment ID                             |
| string | created_at        | Date of creation                       |
| string | updated_at        | Date of updating                       |
| string | method            | Payment method                         |
| string | cc_type           | Credit card type                       |
| string | cc_number_enc     | Credit card number                     |
| string | cc_last4          | Last four digits on the credit card    |
| string | cc_cid_enc        | Credit card CID                        |
| string | cc_owner          | Credit card owner                      |
| string | cc_exp_month      | Credit card expiration month           |
| string | cc_exp_year       | Credit card expiration year            |
| string | cc_ss_owner       | Credit card owner (Switch/Solo)        |
| string | cc_ss_start_month | Credit card start month (Switch/Solo)  |
| string | cc_ss_start_year  | Credit card start year (Switch/Solo)   |
| string | cc_ss_issue       | Credit card issue number (Switch/Solo) |
| string | po_number         | Purchase order number                  |
| string | additional_data   | Additional data                        |

|        |                        |                        |
|--------|------------------------|------------------------|
| string | additional_information | Additional information |
|--------|------------------------|------------------------|

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'cart.info', '15');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartInfo($sessionId, '15');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->shoppingCartInfo((object)array('sessionId' => $sessionId->result,
'quoteId' => '15'));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Cart License

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▲

[Cart](#) ▲

[cart.create](#)

[cart.info](#)

[cart.license](#)

[cart.order](#)

[cart.totals](#)

[Cart Coupon](#) ▾

[Cart Customer](#) ▾

[Cart Payment](#) ▾

[Cart Product](#) ▾

[Cart Shipping](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart

### Method:

- [cart.license](#) (SOAP V1)
- [shoppingCartLicense](#) (SOAP V2)

### Aliases: `cart.license`

Allows you to retrieve the website license agreement for the quote according to the website (store).

### Arguments:

| Type   | Name      | Description                         |
|--------|-----------|-------------------------------------|
| string | sessionId | Session ID                          |
| int    | quotId    | Shopping cart ID (quote identifier) |
| string | store     | Store view ID or code (optional)    |

### Return:

| Type  | Name   | Description                        |
|-------|--------|------------------------------------|
| array | result | Array of shoppingCartLicenseEntity |

The `shoppingCartLicenseEntity` content is as follows:

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

| Type   | Name         | Description                           |
|--------|--------------|---------------------------------------|
| string | agreement_id | License agreement ID                  |
| string | name         | License name                          |
| string | content      | License content                       |
| int    | is_active    | Defines whether the license is active |
| int    | is_html      | Defines whether the license is HTML   |

#### Faults:

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $client->call($session, 'cart.license', '15');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartLicense($sessionId, '15');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->shoppingCartLicense((object)array('sessionId' => $sessionId->result,
'quoteId' => 15));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 0 =>
 array
 'agreement_id' => string '1' (length=1)
 'name' => string 'license' (length=4)
 'content' => string 'terms and conditions' (length=20)
 'content_height' => null
 'checkbox_text' => string 'terms' (length=5)
 'is_active' => string '1' (length=1)
 'is_html' => string '0' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Cart Order

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Cart ▾

cart.create  
cart.info  
cart.license  
**cart.order**  
cart.totals

### Cart Coupon ▾

### Cart Customer ▾

### Cart Payment ▾

### Cart Product ▾

### Cart Shipping ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Mage\_Checkout

### Module: Shopping Cart API

Resource: cart

#### Method:

- cart.order (SOAP V1)
- shoppingCartOrder (SOAP V2)

Allows you to create an order from a shopping cart (quote).

Before placing the order, you need to add the customer, customer address, shipping and payment methods.

#### Arguments:

| Type          | Name      | Description                      |
|---------------|-----------|----------------------------------|
| string        | sessionId | Session ID                       |
| int           | quotId    | Shopping Cart ID (quote ID)      |
| string        | storeId   | Store view ID or code (optional) |
| ArrayOfString | licenses  | Website license ID (optional)    |

#### Return:

| Type   | Name   | Description                 |
|--------|--------|-----------------------------|
| string | result | Result of creating an order |

#### Faults:

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$shoppingCartIncrementId = $proxy->call($sessionId, 'cart.create', array(
'magento_store'));

$resultOrderCreation = $proxy->call(
$sessionId,
"cart.order",
array(
$shoppingCartId
)
);
)
```

### Request Example SOAP V2

```
/**
 * Example of order creation
 * Preconditions are as follows:
 * 1. Create a customer
 * 2. Create a simple product */

$user = 'apiUser';
$password = 'apiKey';
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login($user, $password);
$cartId = $proxy->shoppingCartCreate($sessionId, 1);
// load the customer list and select the first customer from the list
$customerList = $proxy->customerCustomerList($sessionId, array());
$customer = (array) $customerList[0];
$customer['mode'] = 'customer';
$proxy->shoppingCartCustomerSet($sessionId, $cartId, $customer);
// load the product list and select the first product from the list
$productList = $proxy->catalogProductList($sessionId);
$product = (array) $productList[0];
$product['qty'] = 1;
$proxy->shoppingCartProductAdd($sessionId, $cartId, array($product));

$address = array(
array(
'mode' => 'shipping',
'firstname' => $customer['firstname'],
'lastname' => $customer['lastname'],
'street' => 'street address',
'city' => 'city',
'region' => 'region',
'telephone' => 'phone number',
'postcode' => 'postcode',
'country_id' => 'country ID',
'is_default_shipping' => 0,
'is_default_billing' => 0
),
array(
'mode' => 'billing',
'firstname' => $customer['firstname'],
'lastname' => $customer['lastname'],
'street' => 'street address',
)
```

```
 'city' => 'city',
 'region' => 'region',
 'telephone' => 'phone number',
 'postcode' => 'postcode',
 'country_id' => 'country ID',
 'is_default_shipping' => 0,
 'is_default_billing' => 0
),
);
// add customer address
$proxy->shoppingCartCustomerAddresses($sessionId, $cartId, $address);
// add shipping method
$proxy->shoppingCartShippingMethod($sessionId, $cartId, 'flatrate_flatrate');

$paymentMethod = array(
 'po_number' => null,
 'method' => 'checkmo',
 'cc_cid' => null,
 'cc_owner' => null,
 'cc_number' => null,
 'cc_type' => null,
 'cc_exp_year' => null,
 'cc_exp_month' => null
);
// add payment method
$proxy->shoppingCartPaymentMethod($sessionId, $cartId, $paymentMethod);
// place the order
$orderId = $proxy->shoppingCartOrder($sessionId, $cartId, null, null);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Cart Totals

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▲

[Cart](#) ▲

[cart.create](#)

[cart.info](#)

[cart.license](#)

[cart.order](#)

[cart.totals](#)

[Cart Coupon](#) ▾

[Cart Customer](#) ▾

[Cart Payment](#) ▾

[Cart Product](#) ▾

[Cart Shipping](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Mage\_Checkout

### Module: Shopping Cart API

Resource: cart

#### Method:

- [cart.totals](#) (SOAP V1)
- [shoppingCartTotals](#) (SOAP V2)

Allows you to retrieve total prices for a shopping cart (quote).

#### Arguments:

| Type   | Name      | Description                         |
|--------|-----------|-------------------------------------|
| string | sessionId | Session ID                          |
| int    | quotId    | Shopping cart ID (quote identifier) |
| string | store     | Store view ID or code (optional)    |

#### Return:

| Type  | Name   | Description                       |
|-------|--------|-----------------------------------|
| array | result | Array of shoppingCartTotalsEntity |

The [shoppingCartTotalsEntity](#) content is as follows:

| Type   | Name  | Description |
|--------|-------|-------------|
| string | title | Title       |

|       |        |              |
|-------|--------|--------------|
| float | amount | Total amount |
|-------|--------|--------------|

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'cart.totals', '15');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartTotals($sessionId, '15');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->shoppingCartTotals((object)array('sessionId' => $sessionId->result,
'quoteId' => 15));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'title' => string 'Subtotal' (length=8)
 'amount' => float 388.69
1 =>
array
 'title' => string '0 Reward points' (length=15)
 'amount' => float 0
2 =>
array
 'title' => string 'Gift Cards' (length=10)
 'amount' => float 0
3 =>
array
 'title' => string 'Store Credit' (length=12)
 'amount' => float 0
4 =>
array
 'title' => string 'Grand Total' (length=11)
 'amount' => float 388.69
5 =>
```

```
array
 'title' => null
 'amount' => null
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Cart Coupon

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

### Cart ▾

#### Cart Coupon ▾

`cart_coupon.add`  
`cart_coupon.remove`

### Cart Customer ▾

### Cart Payment ▾

### Cart Product ▾

### Cart Shipping ▾

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Enterprise Customer Balance ▾

#### Enterprise Gift Card ▾

#### Enterprise Gift Message ▾

#### Miscellaneous ▾

#### WS-I Compliance

## Module: Mage\_Checkout

The Mage\_Checkout module allows you to manage shopping carts and the checkout process. This module allows you to create an order once filling the shopping cart is complete.

### Cart Coupon

Allows you to add and remove coupon codes for a shopping cart.

**Resource Name:** cart\_coupon

#### Methods:

- [cart\\_coupon.add](#) - Add a coupon code to a quote
- [cart\\_coupon.remove](#) - Remove a coupon code from a quote

**Note:** In Magento, quotes and shopping carts are logically related, but technically different. The shopping cart is a wrapper for a quote, and it is used primarily by the frontend logic. The cart is represented by the Mage\_Checkout\_Model\_Cart class and the quote is represented by the Mage\_Sales\_Model\_Quote class.

### Faults

| Fault Code | Fault Message                                       |
|------------|-----------------------------------------------------|
| 1001       | Can not make operation because store is not exists  |
| 1002       | Can not make operation because quote is not exists  |
| 1081       | Coupon could not be applied because quote is empty. |
| 1082       | Coupon could not be applied.                        |
| 1083       | Coupon is not valid.                                |

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Coupon Add

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Cart ▾

### Cart Coupon ▾

[cart\\_coupon.add](#)

[cart\\_coupon.remove](#)

### Cart Customer ▾

### Cart Payment ▾

### Cart Product ▾

### Cart Shipping ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_coupon

### Method:

- `cart_coupon.add` (SOAP V1)
- `shoppingCartCouponAdd` (SOAP V2)

Allows you to add a coupon code for a shopping cart (quote). The shopping cart must not be empty.

### Arguments:

| Type   | Name       | Description                      |
|--------|------------|----------------------------------|
| string | sessionId  | Session ID                       |
| int    | quotelId   | Shopping cart ID                 |
| string | couponCode | Coupon code                      |
| string | store      | Store view ID or code (optional) |

### Return:

| Type    | Description                      |
|---------|----------------------------------|
| boolean | True if the coupon code is added |

### Faults:

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$shoppingCartId = $proxy->call($sessionId, 'cart.create', array('magento_store'));
$couponCode = "aCouponCode";
$resultCartCouponRemove = $proxy->call(
 $sessionId,
 "cart_coupon.add",
 array(
 $shoppingCartId,
 $couponCode
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartCouponAdd($sessionId, '15', 'aCouponCode');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->shoppingCartCouponAdd((object)array('sessionId' => $sessionId->result, 'quoteId' => 15, 'couponCode' => 'aCouponCode', 'store' => '3'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Coupon Remove

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

[cart\\_coupon.add](#)

[cart\\_coupon.remove](#)

#### Cart Customer ▾

#### Cart Payment ▾

#### Cart Product ▾

#### Cart Shipping ▾

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_coupon

#### Method:

- `cart_coupon.remove` (SOAP V1)
- `shoppingCartCouponRemove` (SOAP V2)

Allows you to remove a coupon code from a shopping cart (quote).

#### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| int    | quotId    | Shopping cart ID                 |
| string | store     | Store view ID or code (optional) |

#### Return:

| Type    | Description                        |
|---------|------------------------------------|
| boolean | True if the coupon code is removed |

#### Faults:

*No Faults.*

## Examples

## Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$shoppingCartId = $proxy->call($sessionId, 'cart.create', array('magento_store'));
$resultCartCouponRemove = $proxy->call(
 $sessionId,
 "cart_coupon.remove",
 array(
 $shoppingCartId
)
);

```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartCouponRemove($sessionId, '15');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->shoppingCartCouponRemove((object)array('sessionId' => $sessionId->result, 'quoteId' => 15, 'store' => '3'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Cart Customer

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

cart\_customer.addresses  
cart\_customer.set

#### Cart Payment ▾

#### Cart Product ▾

#### Cart Shipping ▾

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Checkout

The Mage\_Checkout module allows you to manage shopping carts and the checkout process. This module allows you to create an order once filling the shopping cart is complete.

### Cart Customer

Allows you to add customer information and addresses into a shopping cart.

**Resource Name:** cart\_customer

#### Methods:

- [cart\\_customer.set](#) - Add customer information into a shopping cart
- [cart\\_customer.addresses](#) - Set the customer addresses (shipping and billing) into a shopping cart

## Faults

| Fault Code | Fault Message                                                     |
|------------|-------------------------------------------------------------------|
| 1001       | Can not make operation because store is not exists                |
| 1002       | Can not make operation because quote is not exists                |
| 1041       | Customer is not set.                                              |
| 1042       | The customer's identifier is not valid or customer is not existed |
| 1043       | Customer could not be created.                                    |
| 1044       | Customer data is not valid.                                       |
| 1045       | Customer's mode is unknown                                        |
| 1051       | Customer address data is empty.                                   |

|      |                                                                        |
|------|------------------------------------------------------------------------|
| 1052 | Customer's address data is not valid.                                  |
| 1053 | The customer's address identifier is not valid                         |
| 1054 | Customer address is not set.                                           |
| 1055 | Customer address identifier do not belong customer, which set in quote |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Customer Addresses

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

`cart_customer.addresses`

`cart_customer.set`

#### Cart Payment ▾

#### Cart Product ▾

#### Cart Shipping ▾

### Create Your Own API

#### Customer ▾

#### Directory ▾

#### Sales ▾

#### Enterprise Customer Balance ▾

#### Enterprise Gift Card ▾

#### Enterprise Gift Message ▾

#### Miscellaneous ▾

#### WS-I Compliance

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_customer

#### Method:

- `cart_customer.addresses` (SOAP V1)
- `shoppingCartCustomerAddresses` (SOAP V2)

Allows you to set the customer addresses in the shopping cart (quote).

#### Arguments:

| Type   | Name                | Description                                             |
|--------|---------------------|---------------------------------------------------------|
| string | sessionId           | Session ID                                              |
| int    | quotId              | Shopping cart ID                                        |
| array  | customerAddressData | Array of <code>shoppingCartCustomerAddressEntity</code> |
| string | store               | Store view ID or code (optional)                        |

#### Return:

| Type    | Name   | Description                |
|---------|--------|----------------------------|
| boolean | result | True if the address is set |

The `shoppingCartCustomerAddressEntity` content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |                     |                                                           |
|--------|---------------------|-----------------------------------------------------------|
| string | mode                | Mode: billing or shipping                                 |
| string | address_id          | Address ID                                                |
| string | firstname           | Customer first name                                       |
| string | lastname            | Customer last name                                        |
| string | company             | Company name                                              |
| string | street              | Street                                                    |
| string | city                | City                                                      |
| string | region              | Region                                                    |
| string | region_id           | Region ID                                                 |
| string | postcode            | Post code                                                 |
| string | country_id          | Country ID                                                |
| string | telephone           | Telephone number                                          |
| string | fax                 | Fax number                                                |
| int    | is_default_billing  | Defines whether the address is a default billing address  |
| int    | is_default_shipping | Defines whether the address is a default shipping address |

**Faults:**

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$shoppingCartId = $proxy->call($sessionId, 'cart.create', array('magento_store'));

$arrAddresses = array(
 array(
 "mode" => "shipping",
 "firstname" => "testFirstname",
 "lastname" => "testLastname",
 "company" => "testCompany",
 "street" => "testStreet",
 "city" => "testCity",
 "region" => "testRegion",
 "postcode" => "testPostcode",
 "country_id" => "id",
 "telephone" => "0123456789",
 "fax" => "0123456789",
 "is_default_shipping" => 0,
 "is_default_billing" => 0
),
 array(
 "mode" => "billing",
 "address_id" => "customer_address_id"
)
)
```

```

) ;

$resultCustomerAddresses = $proxy->call(
 $sessionId,
 "cart_customer.addresses",
 array(
 $shoppingCartId,
 $arrAddresses,
)
);

```

## Request Example SOAP V2

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->shoppingCartCustomerAddresses($sessionId, 10, array(array(
'mode' => 'billing',
'firstname' => 'first name',
'lastname' => 'last name',
'street' => 'street address',
'city' => 'city',
'region' => 'region',
'postcode' => 'postcode',
'country_id' => 'US',
'telephone' => '123456789',
'is_default_billing' => 1
)));
var_dump($result);

```

## Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartCustomerAddresses((object)array('sessionId' =>
$sessionId->result, 'quoteId' => 10, 'customerAddressData' => array(array(
'mode' => 'billing',
'firstname' => 'first name',
'lastname' => 'last name',
'street' => 'street address',
'city' => 'city',
'region' => 'region',
'postcode' => 'postcode',
'country_id' => 'US',
'telephone' => '123456789',
'is_default_billing' => 1
)));}
var_dump($result->result);

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Customer Set

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

cart\_customer.addresses

[cart\\_customer.set](#)

#### Cart Payment ▾

#### Cart Product ▾

#### Cart Shipping ▾

### Create Your Own API

#### Customer ▾

#### Directory ▾

#### Sales ▾

#### Enterprise Customer Balance ▾

#### Enterprise Gift Card ▾

#### Enterprise Gift Message ▾

#### Miscellaneous ▾

#### WS-I Compliance

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_customer

#### Method:

- cart\_customer.set (SOAP V1)
- shoppingCartCustomerSet (SOAP V2)

Allows you to add information about the customer to a shopping cart (quote).

#### Arguments:

| Type   | Name         | Description                         |
|--------|--------------|-------------------------------------|
| string | sessionId    | Session ID                          |
| int    | quotId       | Shopping cart ID                    |
| array  | customerData | Array of shoppingCartCustomerEntity |
| string | store        | Store view ID or code (optional)    |

#### Return:

| Type    | Name   | Description                  |
|---------|--------|------------------------------|
| boolean | result | True if information is added |

The **shoppingCartCustomerEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |              |                        |
|--------|--------------|------------------------|
| string | mode         | Customer mode          |
| int    | customer_id  | Customer ID            |
| string | email        | Customer email address |
| string | firstname    | Customer first name    |
| string | lastname     | Customer last name     |
| string | password     | Customer password      |
| string | confirmation | Confirmation flag      |
| int    | website_id   | Website ID             |
| int    | store_id     | Store ID               |
| int    | group_id     | Group ID               |

**Faults:**

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$shoppingCartId = $proxy->call($sessionId, 'cart.create', array('magento_store'));
$customerAsGuest = array(
 "firstname" => "testFirstname",
 "lastname" => "testLastName",
 "email" => "testEmail",
 "website_id" => "0",
 "store_id" => "0",
 "mode" => "guest"
);
$resultCustomerSet = $proxy->call($sessionId, 'cart_customer.set', array(
$shoppingCartId, $customerAsGuest));
```

### Request Example SOAP V2

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$session = $client->login('apiUser', 'apiKey');

$quoteId = $client->shoppingCartCreate($session);

$customerData = array(
 "firstname" => "testFirstname",
 "lastname" => "testLastName",
 "email" => "testEmail@mail.com",
 "mode" => "guest",
 "website_id" => "0"
);

$resultCustomerSet = $client->shoppingCartCustomerSet($session, $quoteId,
$customerData);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Cart Payment

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

### Cart ▾

### Cart Coupon ▾

### Cart Customer ▾

## Cart Payment ▾

`cart_payment.list`

`cart_payment.method`

### Cart Product ▾

### Cart Shipping ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Module: Mage\_Checkout

The Mage\_Checkout module allows you to manage shopping carts and the checkout process. This module allows you to create an order once filling the shopping cart is complete.

### Cart Payment

Allows you to retrieve and set payment methods for a shopping cart.

**Resource Name:** `cart_payment`

#### Methods:

- [`cart\_payment.method`](#) - Set a payment method for a shopping cart
- [`cart\_payment.list`](#) - Get the list of available payment methods for a shopping cart

## Faults

| Fault Code | Fault Message                                                             |
|------------|---------------------------------------------------------------------------|
| 1001       | Can not make operation because store is not exists                        |
| 1002       | Can not make operation because quote is not exists                        |
| 1071       | Payment method data is empty.                                             |
| 1072       | Customer's billing address is not set. Required for payment method data.  |
| 1073       | Customer's shipping address is not set. Required for payment method data. |
| 1074       | Payment method is not allowed                                             |
| 1075       | Payment method is not set.                                                |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Payment List

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

### Cart ▾

### Cart Coupon ▾

### Cart Customer ▾

### Cart Payment ▾

`cart_payment.list`

`cart_payment.method`

### Cart Product ▾

### Cart Shipping ▾

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

## WS-I Compliance

# Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_payment

### Method:

- `cart_payment.list` (SOAP V1)
- `shoppingCartPaymentList` (SOAP V2)

Allows you to retrieve a list of available payment methods for a shopping cart (quote).

### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| int    | quotelId  | Shopping cart ID                 |
| string | store     | Store view ID or code (optional) |

### Return:

| Type  | Name   | Description                                                   |
|-------|--------|---------------------------------------------------------------|
| array | result | Array of <code>shoppingCartPaymentMethodResponseEntity</code> |

The `shoppingCartPaymentMethodResponseEntity` content is as follows:

| Type   | Name | Description         |
|--------|------|---------------------|
| string | code | Payment method code |

|                  |          |                            |
|------------------|----------|----------------------------|
| string           | title    | Payment method title       |
| associativeArray | cc_types | Array of credit card types |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'cart_payment.list', 'quoteId');
var_dump($result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'code' => string 'checkmo' (length=7)
 'title' => string 'Check / Money order' (length=19)
 'ccTypes' => null
1 =>
array
 'code' => string 'ccsave' (length=6)
 'title' => string 'Credit Card (saved)' (length=19)
 'ccTypes' =>
 array
 'AE' => string 'American Express' (length=16)
 'VI' => string 'Visa' (length=4)
 'MC' => string 'MasterCard' (length=10)
 'DI' => string 'Discover' (length=8)
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Payment Method

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▲

[Cart](#) ▾

[Cart Coupon](#) ▾

[Cart Customer](#) ▾

[Cart Payment](#) ▲

[cart\\_payment.list](#)

[cart\\_payment.method](#)

[Cart Product](#) ▾

[Cart Shipping](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_payment

### Method:

- [cart\\_payment.method \(SOAP V1\)](#)
- [shoppingCartPaymentMethod \(SOAP V2\)](#)

Allows you to set a payment method for a shopping cart (quote).

### Arguments:

| Type   | Name      | Description                              |
|--------|-----------|------------------------------------------|
| string | sessionId | Session ID                               |
| int    | quotId    | Shopping cart ID                         |
| array  | method    | Array of shoppingCartPaymentMethodEntity |
| string | store     | Store view ID or code (optional)         |

### Return:

| Type    | Description     |
|---------|-----------------|
| boolean | True on success |

The **shoppingCartPaymentMethodEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |              |                              |
|--------|--------------|------------------------------|
| string | po_number    | Purchase order number        |
| string | method       | Payment method               |
| string | cc_cid       | Credit card CID              |
| string | cc_owner     | Credit card owner            |
| string | cc_number    | Credit card number           |
| string | cc_type      | Credit card type             |
| string | cc_exp_year  | Credit card expiration year  |
| string | cc_exp_month | Credit card expiration month |

**Faults:**

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$shoppingCartId = $proxy->call($sessionId, 'cart.create', array('magento_store'));

$paymentMethod = array(
 "method" => "checkmo"
);

$resultPaymentMethod = $proxy->call(
 $sessionId,
 "cart_payment.method",
 array(
 $shoppingCartId,
 $paymentMethod
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->shoppingCartPaymentMethod($sessionId, 10, array(
 'po_number' => null,
 'method' => 'checkmo',
 'cc_cid' => null,
 'cc_owner' => null,
 'cc_number' => null,
 'cc_type' => null,
 'cc_exp_year' => null,
 'cc_exp_month' => null
));

var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartPaymentMethod((object)array('sessionId' => $sessionId-
>result, 'quoteId' => 10, 'method' => array(
'po_number' => null,
'method' => 'checkmo',
'cc_cid' => null,
'cc_owner' => null,
'cc_number' => null,
'cc_type' => null,
'cc_exp_year' => null,
'cc_exp_month' => null
)));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Cart Product

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

#### Cart Payment ▾

### Cart Product ▾

cart\_product.add  
cart\_product.list  
cart\_product.moveToCustomer...  
cart\_product.remove  
cart\_product.update

#### Cart Shipping ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Mage\_Checkout

The Mage\_Checkout module allows you to manage shopping carts and the checkout process. This module allows you to create an order once filling the shopping cart is complete.

### Cart Product

Allows you to manage products in a shopping cart.

**Resource Name:** cart\_product

#### Methods:

- [cart\\_product.add](#) - Add one or more products to a shopping cart
- [cart\\_product.update](#) - Update one or more products in a shopping cart
- [cart\\_product.remove](#) - Remove one or more products from a shopping cart
- [cart\\_product.list](#) - Get a list of products in a shopping cart
- [cart\\_product.moveToCustomerQuote](#) - Move one or more products from the quote to the customer shopping cart

## Faults

| Fault Code | Fault Message                                                |
|------------|--------------------------------------------------------------|
| 1001       | Can not make operation because store is not exists           |
| 1002       | Can not make operation because quote is not exists           |
| 1021       | Product's data is not valid.                                 |
| 1022       | Product(s) could not be added.                               |
| 1023       | Quote could not be saved during adding product(s) operation. |
| 1024       | Product(s) could not be updated.                             |

|      |                                                                     |
|------|---------------------------------------------------------------------|
| 1025 | Quote could not be saved during updating product(s) operation.      |
| 1026 | Product(s) could not be removed.                                    |
| 1027 | Quote could not be saved during removing product(s) operation.      |
| 1028 | Customer is not set for quote.                                      |
| 1029 | Customer's quote is not existed.                                    |
| 1030 | Quotes are identical.                                               |
| 1031 | Product(s) could not be moved.                                      |
| 1032 | One of quote could not be saved during moving product(s) operation. |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Add

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

#### Catalog Inventory ▾

#### Checkout ▾

##### Cart ▾

##### Cart Coupon ▾

##### Cart Customer ▾

##### Cart Payment ▾

##### Cart Product ▾

`cart_product.add`

`cart_product.list`

`cart_product.moveToCustomer...`

`cart_product.remove`

`cart_product.update`

##### Cart Shipping ▾

### Create Your Own API

#### Customer ▾

#### Directory ▾

#### Sales ▾

#### Enterprise Customer Balance ▾

#### Enterprise Gift Card ▾

#### Enterprise Gift Message ▾

#### Miscellaneous ▾

#### WS-I Compliance

## Shopping Cart API

Allows you to create/modify shopping cart and create an order after complete filling the shopping cart. Consists of two main parts: Shopping Cart and Checkout processes.

**Module:** Mage\_Checkout

**Resource:** cart\_product

### Method:

- `cart_product.add` (SOAP V1)
- `shoppingCartProductAdd` (SOAP V2)

Allows you to add one or more products to the shopping cart (quote).

### Arguments:

| Type   | Name                  | Description                                         |
|--------|-----------------------|-----------------------------------------------------|
| string | sessionId             | Session ID                                          |
| int    | quotId                | Shopping cart ID (quote ID)                         |
| array  | products productsData | An array with the list of shoppingCartProductEntity |
| string | storeId               | Store view ID or code (optional)                    |

### Returns:

| Type    | Description                                                    |
|---------|----------------------------------------------------------------|
| boolean | True on success (if the product is added to the shopping cart) |

The shoppingCartProductEntity array attributes are as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|                  |                   |                                                                        |
|------------------|-------------------|------------------------------------------------------------------------|
| string           | product_id        | ID of the product to be added to the shopping cart (quote) (optional)  |
| string           | sku               | SKU of the product to be added to the shopping cart (quote) (optional) |
| double           | qty               | Number of products to be added to the shopping cart (quote) (optional) |
| associativeArray | options           | An array in the form of option_id => content (optional)                |
| associativeArray | bundle_option     | An array of bundle item options (optional)                             |
| associativeArray | bundle_option_qty | An array of bundle items quantity (optional)                           |
| ArrayOfString    | links             | An array of links (optional)                                           |

**Faults:**

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$quoteId = $proxy->call($sessionId, 'cart.create', array('magento_store'));
$arrProducts = array(
 array(
 "product_id" => "1",
 "qty" => 2
 "options" => array(
 optionId_1 => optionValue_1,
 ...,
 optionId_n => optionValue_n
)
),
 array(
 "sku" => "testSKU",
 "quantity" => 4
)
);
$resultCartProductAdd = $proxy->call(
 $sessionId,
 "cart_product.add",
 array(
 $quoteId,
 $arrProducts
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->shoppingCartProductAdd($sessionId, 10, array(array(
'product_id' => '4',
'sku' => 'simple_product',
'qty' => '5',
'options' => null,
```

```
'bundle_option' => null,
'bundle_option_qty' => null,
'links' => null
));

var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartProductAdd((object)array('sessionId' => $sessionId-
>result, 'quoteId' => 10, 'productsData' => array(array(
'product_id' => '4',
'sku' => 'simple_product',
'qty' => '1',
'options' => null,
'bundle_option' => null,
'bundle_option_qty' => null,
'links' => null
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

#### Cart Payment ▾

#### Cart Product ▾

cart\_product.add

[cart\\_product.list](#)

cart\_product.moveToCustomer...

cart\_product.remove

cart\_product.update

#### Cart Shipping ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_product

#### Method:

- cart\_product.list (SOAP V1)
- shoppingCartProductList (SOAP V2)

Allows you to retrieve the list of products in the shopping cart (quote).

#### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| int    | quotId    | Shopping cart ID                 |
| string | store     | Store view ID or code (optional) |

#### Return:

| Type  | Name   | Description                                |
|-------|--------|--------------------------------------------|
| array | result | Array of shoppingCartProductResponseEntity |

The **shoppingCartProductResponseEntity** (**catalogProductEntity**) content is as follows:

| Type   | Name       | Description |
|--------|------------|-------------|
| string | product_id | Product ID  |

|               |              |                       |
|---------------|--------------|-----------------------|
| string        | sku          | Product SKU           |
| string        | name         | Product name          |
| string        | set          | Product attribute set |
| string        | type         | Product type          |
| ArrayOfString | category_ids | Array of category IDs |
| ArrayOfString | website_ids  | Array of website IDs  |

#### Faults:

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'cart_product.list', '15');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartProductList($sessionId, '15');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartProductList((object)array('sessionId' => $sessionId-
>result, 'quoteId' => 15));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
'product_id' => string '3' (length=1)
'sku' => string 'canonxt' (length=7)
'name' => string 'Canon Digital Rebel XT 8MP Digital SLR Camera' (length=45)
'set' => string '4' (length=1)
'type' => string 'simple' (length=6)
'category_ids' =>
array
```

```
 0 => string '5' (length=1)
'website_ids' =>
array
 0 => string '2' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Move To Customer Quote

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

#### Cart Payment ▾

#### Cart Product ▾

cart\_product.add

cart\_product.list

[cart\\_product.moveToCustomer...](#)

cart\_product.remove

cart\_product.update

#### Cart Shipping ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_product

#### Method:

- cart\_product.moveToCustomerQuote (SOAP V1)
- shoppingCartProductMoveToCustomerQuote (SOAP V2)

Allows you to move products from the current quote to a customer quote.

#### Arguments:

| Type   | Name         | Description                        |
|--------|--------------|------------------------------------|
| string | sessionId    | Session ID                         |
| int    | quotId       | Shopping cart ID                   |
| array  | productsData | Array of shoppingCartProductEntity |
| string | store        | Store view ID or code (optional)   |

#### Return:

| Type    | Name   | Description                                    |
|---------|--------|------------------------------------------------|
| boolean | result | True if the product is moved to customer quote |

The **shoppingCartProductEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|                  |                   |                                              |
|------------------|-------------------|----------------------------------------------|
| string           | product_id        | Product ID                                   |
| string           | sku               | Product SKU                                  |
| double           | qty               | Product quantity                             |
| associativeArray | options           | Product custom options                       |
| associativeArray | bundle_option     | An array of bundle item options (optional)   |
| associativeArray | bundle_option_qty | An array of bundle items quantity (optional) |
| ArrayOfString    | links             | An array of links (optional)                 |

**Faults:**

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$shoppingCartIncrementId = $proxy->call($sessionId, 'cart.create', array(
'magento_store'));
$arrProducts = array(
 array(
 "product_id" => "1",
 "qty" => 2
),
 array(
 "sku" => "testSKU",
 "quantity" => 4
)
);
$resultCartProductAdd = $proxy->call(
 $sessionId,
 "cart_product.add",
 array(
 $shoppingCartId,
 $arrProducts
)
);
$arrProducts = array(
 array(
 "product_id" => "1"
),
);
$resultCartProductMove = $proxy->call(
 $sessionId,
 "cart_product.moveToCustomerQuote",
 array(
 $shoppingCartId,
 $arrProducts
)
);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Product Remove

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

### Cart ▾

### Cart Coupon ▾

### Cart Customer ▾

### Cart Payment ▾

### Cart Product ▾

cart\_product.add

cart\_product.list

cart\_product.moveToCustomer...

[cart\\_product.remove](#)

cart\_product.update

### Cart Shipping ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

# Mage\_Checkout

## Module: Shopping Cart API

### Resource: cart\_product

#### Method:

- cart\_product.remove (SOAP V1)
- shoppingCartProductRemove (SOAP V2)

Allows you to remove one or several products from a shopping cart (quote).

#### Arguments:

| Type   | Name         | Description                        |
|--------|--------------|------------------------------------|
| string | sessionId    | Session ID                         |
| int    | quotId       | Shopping cart ID                   |
| array  | productsData | Array of shoppingCartProductEntity |
| string | store        | Store view ID or code (optional)   |

#### Return:

| Type    | Description                    |
|---------|--------------------------------|
| boolean | True if the product is removed |

The **shoppingCartProductEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|                  |                   |                                              |
|------------------|-------------------|----------------------------------------------|
| string           | product_id        | Product ID                                   |
| string           | sku               | Product SKU                                  |
| double           | qty               | Product quantity                             |
| associativeArray | options           | Product custom options                       |
| associativeArray | bundle_option     | An array of bundle item options (optional)   |
| associativeArray | bundle_option_qty | An array of bundle items quantity (optional) |
| ArrayOfString    | links             | An array of links (optional)                 |

**Faults:**

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$shoppingCartIncrementId = $proxy->call($sessionId, 'cart.create', array(
'magento_store'));
$arrProducts = array(
 array(
 "product_id" => "1",
 "qty" => 2
),
 array(
 "sku" => "testSKU",
 "quantity" => 4
)
);
$resultCartProductAdd = $proxy->call(
 $sessionId,
 "cart_product.add",
 array(
 $shoppingCartId,
 $arrProducts
)
);
$arrProducts = array(
 array(
 "product_id" => "1"
),
);
$resultCartProductUpdate = $proxy->call(
 $sessionId,
 "cart_product.remove",
 array(
 $shoppingCartId,
 $arrProducts
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
```

```

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->shoppingCartProductRemove($sessionId, 10, array(array(
 'product_id' => '4',
 'sku' => 'simple_product',
 'qty' => '1',
 'options' => null,
 'bundle_option' => null,
 'bundle_option_qty' => null,
 'links' => null
))));

var_dump($result);

```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartProductRemove((object)array('sessionId' => $sessionId-
>result, 'quoteId' => 10, 'productsData' => array(array(
 'product_id' => '4',
 'sku' => 'simple_product',
 'qty' => '1',
 'options' => null,
 'bundle_option' => null,
 'bundle_option_qty' => null,
 'links' => null
))));

var_dump($result->result);

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Product Update

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

#### Cart Payment ▾

#### Cart Product ▾

`cart_product.add`

`cart_product.list`

`cart_product.moveToCustomer...`

`cart_product.remove`

`cart_product.update`

#### Cart Shipping ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_product

#### Method:

- `cart_product.update` (SOAP V1)
- `shoppingCartProductUpdate` (SOAP V2)

Allows you to update one or several products in the shopping cart (quote).

#### Arguments:

| Type   | Name         | Description                        |
|--------|--------------|------------------------------------|
| string | sessionId    | Session ID                         |
| int    | quotId       | Shopping cart ID                   |
| array  | productsData | Array of shoppingCartProductEntity |
| string | store        | Store view ID or code (optional)   |

#### Return:

| Type    | Description                    |
|---------|--------------------------------|
| boolean | True if the product is updated |

The `shoppingCartProductEntity` content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|                  |                   |                                              |
|------------------|-------------------|----------------------------------------------|
| string           | product_id        | Product ID                                   |
| string           | sku               | Product SKU                                  |
| double           | qty               | Product quantity                             |
| associativeArray | options           | Product custom options                       |
| associativeArray | bundle_option     | An array of bundle item options (optional)   |
| associativeArray | bundle_option_qty | An array of bundle items quantity (optional) |
| ArrayOfString    | links             | An array of links (optional)                 |

**Faults:**

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$shoppingCartIncrementId = $proxy->call($sessionId, 'cart.create', array(
'magento_store'));
$arrProducts = array(
 array(
 "product_id" => "1",
 "qty" => 2
),
 array(
 "sku" => "testSKU",
 "quantity" => 4
)
);
$resultCartProductAdd = $proxy->call(
 $sessionId,
 "cart_product.add",
 array(
 $shoppingCartId,
 $arrProducts
)
);
$arrProducts = array(
 array(
 "product_id" => "1",
 "qty" => 5
),
);
$resultCartProductUpdate = $proxy->call(
 $sessionId,
 "cart_product.update",
 array(
 $shoppingCartId,
 $arrProducts
)
);
```

### Request Example SOAP V2

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->shoppingCartProductUpdate($sessionId, 10, array(array(
 'product_id' => '4',
 'sku' => 'simple_product',
 'qty' => '2',
 'options' => null,
 'bundle_option' => null,
 'bundle_option_qty' => null,
 'links' => null
)));

var_dump($result);

```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartProductUpdate((object)array('sessionId' => $sessionId-
>result, 'quoteId' => 10, 'productsData' => array(array(
 'product_id' => '4',
 'sku' => 'simple_product',
 'qty' => '5',
 'options' => null,
 'bundle_option' => null,
 'bundle_option_qty' => null,
 'links' => null
))));

var_dump($result->result);

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Cart Shipping

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

#### Cart Payment ▾

#### Cart Product ▾

### Cart Shipping ▾

`cart_shipping.list`

`cart_shipping.method`

## Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Checkout

The Mage\_Checkout module allows you to manage shopping carts and the checkout process. This module allows you to create an order once filling the shopping cart is complete.

### Cart Shipping

Allows you to retrieve and set shipping methods for a shopping cart.

**Resource Name:** `cart_shipping`

#### Methods:

- `cart_shipping.method` - Set a shipping method for a shopping cart
- `cart_shipping.list` - Retrieve the list of available shipping methods for a shopping cart

## Faults

| Fault Code | Fault Message                                                          |
|------------|------------------------------------------------------------------------|
| 1001       | Can not make operation because store is not exists                     |
| 1002       | Can not make operation because quote is not exists                     |
| 1061       | Can not make operation because of customer shipping address is not set |
| 1062       | Shipping method is not available                                       |
| 1063       | Can not set shipping method.                                           |
| 1064       | Can not receive list of shipping methods.                              |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Shipping List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

#### Cart Payment ▾

#### Cart Product ▾

#### Cart Shipping ▾

[cart\\_shipping.list](#)

[cart\\_shipping.method](#)

### Create Your Own API

#### Customer ▾

#### Directory ▾

#### Sales ▾

#### Enterprise Customer Balance ▾

#### Enterprise Gift Card ▾

#### Enterprise Gift Message ▾

#### Miscellaneous ▾

#### WS-I Compliance

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_shipping

#### Method:

- `cart_shipping.list` (SOAP V1)
- `shoppingCartShippingList` (SOAP V2)

Allows you to retrieve the list of available shipping methods for a shopping cart (quote).

#### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| int    | quotId    | Shopping cart ID                 |
| string | storeId   | Store view ID or code (optional) |

#### Returns:

| Type  | Name   | Description                                            |
|-------|--------|--------------------------------------------------------|
| array | result | Array of <code>shoppingCartShippingMethodEntity</code> |

The `shoppingCartShippingMethodEntity` content is as follows:

| Type   | Name | Description |
|--------|------|-------------|
| string | code | Code        |

|        |                    |                             |
|--------|--------------------|-----------------------------|
| string | carrier            | Carrier                     |
| string | carrier_title      | Carrier title               |
| string | method             | Shipping method             |
| string | method_title       | Shipping method title       |
| string | method_description | Shipping method description |
| double | price              | Shipping price              |

**Faults:**

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->call($sessionId, 'cart_shipping.list', 10);

var_dump($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->shoppingCartShippingList($sessionId, 10);

var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartShippingList((object)array('sessionId' => $sessionId-
>result, 'quoteId' => 10));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Shipping Method

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

#### Cart ▾

#### Cart Coupon ▾

#### Cart Customer ▾

#### Cart Payment ▾

#### Cart Product ▾

#### Cart Shipping ▾

[cart\\_shipping.list](#)

[cart\\_shipping.method](#)

### Create Your Own API

#### Customer ▾

#### Directory ▾

#### Sales ▾

#### Enterprise Customer Balance ▾

#### Enterprise Gift Card ▾

#### Enterprise Gift Message ▾

#### Miscellaneous ▾

#### WS-I Compliance

## Mage\_Checkout

Module: Shopping Cart API

Resource: cart\_shipping

#### Method:

- `cart_shipping.method` (SOAP V1)
- `shoppingCartShippingMethod` (SOAP V2)

Allows you to set a shipping method for a shopping cart (quote).

#### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| int    | quotId    | Shopping cart ID                 |
| string | method    | Shipping method code             |
| string | storeId   | Store view ID or code (optional) |

#### Return:

| Type    | Name   | Description                        |
|---------|--------|------------------------------------|
| boolean | result | True if the shipping method is set |

#### Faults:

*No Faults.*

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->call($sessionId, 'cart_shipping.method', array(10,
'freeshipping_freeshipping'));

var_dump($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->shoppingCartShippingMethod($sessionId, 10,
'freeshipping_freeshipping');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartShippingMethod((object)array('sessionId' => $sessionId-
>result, 'quoteId' => 10, 'shippingMethod' => 'freeshipping_freeshipping'));
var_dump($result->result);
```

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Create Your Own API

[Edit this page on GitHub](#) 

## Introduction

## Catalog

## Catalog Inventory

## Checkout

## Create Your Own API

## Customer

## Directory

## Sales

## Enterprise Customer Balance

## Enterprise Gift Card

## Enterprise Gift Message

## Miscellaneous

## WS-I Compliance

- [Creating a Custom API or Extending the Core API](#)
  - [1. Creating an XML File that Will Define the API Resource](#)
  - [2. Adding a Resource Named Customer](#)
  - [3. Adding Faults](#)
  - [4. Describing the Access Control List \(ACL\) for the Resource](#)
  - [5. Creating PHP Code](#)
  - [Creating a Custom Adapter](#)
  - [Common Error Messages](#)

## Creating a Custom API or Extending the Core API

The Core API allows you to manage a set of common resources used in Magento. However, you may choose to have your own set of resources to manage, or you may wish to extend the Core API to handle additional resources.

This tutorial leads you through the process of creating a custom API for a customer module that handles basic customer information.

**Note:** This tutorial applies to v1 of the API.

To learn more about the Core API, to read Magento Core API calls.

For general information about the Magento API, go to the Introduction.

### 1. Creating an XML File that Will Define the API Resource

Create a file named **api.xml** in the /etc folder in the customer module. Start with the empty structure, as follows:

```
<config>
 <api>
 <resources>
```

```

 </resources>
 <acl>
 <resources>
 <all>
 </all>
 </resources>
 </acl>
 </api>
</config>

```

## 2. Adding a Resource Named Customer

Add an element named customer in the `<resources>` element. Add a `<methods>` element, with elements for list, create, info, update and remove methods for customer resource.

Note that:

- list will return all customers
- create will create a new customer
- info will return data on a specified customer
- update will update data on a specified customer
- remove will delete data on a specified customer

```

<config>
 <api>

 <resources>
 <customer translate="title" module="customer">
 <title>Customer Resource</title>
 <methods>
 <list translate="title" module="customer">
 <title>Retrieve customers</title>
 </list>
 <create translate="title" module="customer">
 <title>Create customer</title>
 </create>
 <info translate="title" module="customer">
 <title>Retrieve customer data</title>
 </info>
 <update translate="title" module="customer">
 <title>Update customer data</title>
 </update>
 <delete>
 <title>Delete customer</title>
 </delete>
 </methods>
 <faults module="customer">
 </faults>
 </customer>
 </resources>

 </api>
</config>

```

## 3. Adding Faults

The resource can return some faults, so add a `<faults>` element in the customer element, and list the various faults.

```

<config>

```

```

<api>
 ...
 <resources>
 <customer translate="title" module="customer">
 ...
 <faults module="customer"> <!-- module="customer" specifies the module which will be used for translation. -->
 <data_invalid> <!-- if we get invalid input data for customers -->
 <code>100</code >
 <!-- we cannot know all the errors that can appear, their details can be found in error message for call -->
 <message>Invalid customer data. Details in error message.
 </message>
 </data_invalid>
 <filters_invalid>
 <code>101</code >
 <message>Invalid filters specified. Details in error message.
 </filters_invalid>
 <not_exists>
 <code>102</code >
 <message>Customer doesn't exist.</message>
 </not_exists>
 <not_deleted>
 <code>103</code >
 <message>Customer was not deleted. Details in error message.
 </not_deleted>
 </faults>
 </customer>
 </resources>
 ...
</api>
</config>

```

## 4. Describing the Access Control List (ACL) for the Resource

In order to prevent unauthorized access to our custom API, you must first list the resources that are restricted within the `<acl>` element.

```

<config>
 <api>
 ...
 <acl>
 <resources>
 <customer translate="title" module="customer">
 <title>Customers</title>
 <list translate="title" module="customer">
 <title>View All</title>
 </list>
 <create translate="title" module="customer">
 <title>Create</title>
 </create>
 <info translate="title" module="customer">
 <title>Get Info</title>
 </info>
 <update translate="title" module="customer">
 <title>Update</title>
 </update>
 <delete translate="title" module="customer">
 <title>Delete</title>
 </delete>
 </customer>
 </resources>
 </acl>
 </api>
 </config>

```

```

 </customer>
 </resources>
</acl>
</api>
</config>
```

Then, map ACL resources to API resource methods by adding an `<acl>` element to each part of the resource that needs restricting:

```

<config>
 <api>
 <resources>
 <customer translate="title" module="customer">
 <title>Customer Resource</title>
 <acl>customer</acl>
 <methods>
 <list translate="title" module="customer">
 <title>Retrieve customers</title>
 <acl>customer/list</acl>
 </list>
 <create translate="title" module="customer">
 <title>Create customer</title>
 <acl>customer/create</acl>
 </create>
 <info translate="title" module="customer">
 <title>Retrieve customer data</title>
 <acl>customer/info</acl>
 </info>
 <update translate="title" module="customer">
 <title>Update customer data</title>
 <acl>customer/update</acl>
 </update>
 <delete>
 <title>Delete customer</title>
 <acl>customer/delete</acl>
 </delete>
 </methods>

 </customer>
 </resources>

 </api>
</config>
```

## 5. Creating PHP Code

Next, write some PHP code to access the resources. Start by creating a class called `Mage_Customer_Model_Customer_Api` that extends `Mage_Api_Model_Resource_Abstract`. Save it into a file called **api.php**.

```

class Mage_Customer_Model_Customer_Api extends Mage_Api_Model_Resource_Abstract
{

 public function create($customerData)
 {
 }

 public function info($customerId)
 {
 }
```

```

public function items($filters)
{
}

public function update($customerId, $customerData)
{
}

public function delete($customerId)
{
}

```

Note that you cannot create method "list" because it's a PHP keyword, so instead the method is named items. In order to make this work, add a <method> element into the <list> element in **api.xml**, as shown below.

```

<config>
 <api>
 <resources>
 <customer translate="title" module="customer">
 <model>customer/api</model> <!-- our model -->
 <title>Customer Resource</title>
 <acl>customer</acl>
 <methods>
 <list translate="title" module="customer">
 <title>Retrieve customers</title>
 <method>items</method> <!-- we have another method name inside
our resource -->
 <acl>customer/list</acl>
 </list>

 </methods>

 </resources>

 </api>
 </config>

```

Now add some simple functionality to the Mage\_Customer\_Model\_Api methods you created.

Create a customer:

```

public function create($customerData)
{
 try {
 $customer = Mage::getModel('customer/customer')
 ->setData($customerData)
 ->save();
 } catch (Mage_Core_Exception $e) {
 $this->_fault('data_invalid', $e->getMessage());
 // We cannot know all the possible exceptions,
 // so let's try to catch the ones that extend Mage_Core_Exception
 } catch (Exception $e) {
 $this->_fault('data_invalid', $e->getMessage());
 }
 return $customer->getId();
}

```

Retrieve customer info:

```

public function info($customerId)

```

```

{
 $customer = Mage::getModel('customer/customer')->load($customerId);
 if (!$customer->getId()) {
 $this->_fault('not_exists');
 // If customer not found.
 }
 return $customer->toArray();
 // We can use only simple PHP data types in webservices.
}

```

Retrieve list of customers using filtering:

```

public function items($filters)
{
 $collection = Mage::getModel('customer/customer')->getCollection()
 ->addAttributeToSelect('*');

 if (is_array($filters)) {
 try {
 foreach ($filters as $field => $value) {
 $collection->addFieldToFilter($field, $value);
 }
 } catch (Mage_Core_Exception $e) {
 $this->_fault('filters_invalid', $e->getMessage());
 // If we are adding filter on non-existent attribute
 }
 }

 $result = array();
 foreach ($collection as $customer) {
 $result[] = $customer->toArray();
 }

 return $result;
}

```

Update a customer:

```

public function update($customerId, $customerData)
{
 $customer = Mage::getModel('customer/customer')->load($customerId);

 if (!$customer->getId()) {
 $this->_fault('not_exists');
 // No customer found
 }

 $customer->addData($customerData)->save();
 return true;
}

```

Delete a customer:

```

public function delete($customerId)
{
 $customer = Mage::getModel('customer/customer')->load($customerId);

 if (!$customer->getId()) {
 $this->_fault('not_exists');
 // No customer found
 }
}

```

```

 try {
 $customer->delete();
 } catch (Mage_Core_Exception $e) {
 $this->_fault('not_deleted', $e->getMessage());
 // Some errors while deleting.
 }

 return true;
 }
}

```

## Creating a Custom Adapter

In order to create custom webservice adapter, implement the Mage\_Api\_Model\_Server\_Adapter\_Interface, which is shown below.

```

interface Mage_Api_Model_Server_Adapter_Interface
{
 /**
 * Set handler class name for webservice
 *
 * @param string $handler
 * @return Mage_Api_Model_Server_Adapter_Interface
 */
 function setHandler($handler);

 /**
 * Retrieve handler class name for webservice
 *
 * @return string [basc]
 */
 function getHandler();

 /**
 * Set webservice API controller
 *
 * @param Mage_Api_Controller_Action $controller
 * @return Mage_Api_Model_Server_Adapter_Interface
 */
 function setController(Mage_Api_Controller_Action $controller);

 /**
 * Retrieve webservice API controller
 *
 * @return Mage_Api_Controller_Action
 */
 function getController();

 /**
 * Run webservice
 *
 * @return Mage_Api_Model_Server_Adapter_Interface
 */
 function run();

 /**
 * Dispatch webservice fault
 *
 * @param int $code
 * @param string $message
 */
}

```

```

 function fault($code, $message);

 } // Class Mage_Api_Model_Server_Adapter_Interface End

```

Here is an example implementation for XML-RPC:

```

class Mage_Api_Model_Server_Adapter_Customxmlrpc
 extends Varien_Object
 implements Mage_Api_Model_Server_Adapter_Interface
{
 /**
 * XmlRpc Server
 *
 * @var Zend_XmlRpc_Server
 */
 protected $_xmlRpc = null;

 /**
 * Set handler class name for webservice
 *
 * @param string $handler
 * @return Mage_Api_Model_Server_Adapter_Xmlrpc
 */
 public function setHandler($handler)
 {
 $this->setData('handler', $handler);
 return $this;
 }

 /**
 * Retrieve handler class name for webservice
 *
 * @return string
 */
 public function getHandler()
 {
 return $this->getData('handler');
 }

 /**
 * Set webservice API controller
 *
 * @param Mage_Api_Controller_Action $controller
 * @return Mage_Api_Model_Server_Adapter_Xmlrpc
 */
 public function setController(Mage_Api_Controller_Action $controller)
 {
 $this->setData('controller', $controller);
 return $this;
 }

 /**
 * Retrieve webservice API controller
 *
 * @return Mage_Api_Controller_Action
 */
 public function getController()
 {
 return $this->getData('controller');
 }

 /**
 * Run webservice
 */
}

```

```

*
* @param Mage_Api_Controller_Action $controller
* @return Mage_Api_Model_Server_Adapter_Xmlrpc
*/
public function run()
{
 $this->_xmlRpc = new Zend_XmlRpc_Server();
 $this->_xmlRpc->setClass($this->getHandler());
 $this->getController()->getResponse()
 ->setHeader('Content-Type', 'text/xml')
 ->setBody($this->_xmlRpc->handle());
 return $this;
}

/**
 * Dispatch webservice fault
 *
 * @param int $code
 * @param string $message
 */
public function fault($code, $message)
{
 throw new Zend_XmlRpc_Server_Exception($message, $code);
}
} // Class Mage_Api_Model_Server_Adapter_Customxmlrpc End

```

**Notes:** The setHandler, getHandler, setController and getController methods have a simple implementation that uses the Varien\_Object getData and setData methods.

The run and fault methods are a native implementation for an XML-RPC webservice. The run method defines webservice logic in this adapter for creating an XML-RPC server to handle XML-RPC requests.

```

public function run()
{
 $this->_xmlRpc = new Zend_XmlRpc_Server();
 $this->_xmlRpc->setClass($this->getHandler());
 $this->getController()->getResponse()
 ->setHeader('Content-Type', 'text/xml')
 ->setBody($this->_xmlRpc->handle());
 return $this;
}

```

The "fault" method allows you to send fault exceptions for XML-RPC service when handling requests.

```

public function fault($code, $message)
{
 throw new Zend_XmlRpc_Server_Exception($message, $code);
}

```

## Common Error Messages

The following are common error messages that you might receive when creating your own custom API.

### Invalid API path

This error occurs when the methods listed in the **api.xml** file do not correspond exactly with those used in your PHP file.

For example, in your api.xml file, you might have this:

```
<config>
 <api>
 <resources>
 <checkout_cart translate="title" module="checkout">
 <model>checkout/cart_api</model>
 <title>Cart API</title>
 <methods>
 <list translate="title" module="checkout">
 <title>Retrieve cart data</title>
 <method>info</method>
 </list>
 </methods>
 </checkout_cart>
 </resources>
 ...
 </api>
</config>
```

You should have a corresponding info method in your PHP file.

```
class Mage_Checkout_Model_Cart_Api extends Mage_Cart_Model_Api_Resource
{
 public function info()
 {
 ...
 }
}
```

If you are missing this method, the error "Invalid API path" will be returned.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Customer

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

customer\_group

customer.create

customer.delete

customer.info

customer.list

customer.update

## Customer Address ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Module: Mage\_Customer

The Mage\_Customer module allows you to create, retrieve, update, and delete customers and customer addresses.

### Customer

Allows you to create, retrieve, update, and delete data about customers.

**Resource Name:** customer

#### Methods:

- [customer.list](#) - Retrieve the list of customers
- [customer.create](#) - Create a new customer
- [customer.info](#) - Retrieve the customer data
- [customer.update](#) - Update the customer data
- [customer.delete](#) - Delete a required customer

### Faults

| Fault Code | Fault Message                                        |
|------------|------------------------------------------------------|
| 100        | Invalid customer data. Details in error message.     |
| 101        | Invalid filters specified. Details in error message. |
| 102        | Customer does not exist.                             |
| 103        | Customer not deleted. Details in error message.      |

### Examples

## Example 1. View, create, update, and delete a customer

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');

$newCustomer = array(
 'firstname' => 'First',
 'lastname' => 'Last',
 'email' => 'test@example.com',
 //for my version of magento (1.3.2.4) you SHOULD NOT
 // hash the password, as in:
 // 'password_hash' => 'password'
 'password_hash' => md5('password'),
 // password hash can be either regular or salted md5:
 // $hash = md5($password);
 // $hash = md5($salt.$password).':'.$salt;
 // both variants are valid
 'store_id' => 0,
 'website_id' => 0
);

$newCustomerId = $proxy->call($sessionId, 'customer.create', array($newCustomer));

// Get new customer info
var_dump($proxy->call($sessionId, 'customer.info', $newCustomerId));

// Update customer
$update = array('firstname'=>'Changed Firstname');
$proxy->call($sessionId, 'customer.update', array($newCustomerId, $update));

var_dump($proxy->call($sessionId, 'customer.info', $newCustomerId));

// Delete customer
$proxy->call($sessionId, 'customer.delete', $newCustomerId);
```

## Customer Groups

Allows you to retrieve the customer groups.

**Resource Name:** customer\_group

**Methods:**

- [customer\\_group.list](#) - Retrieve the list of customer groups

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Customer Group

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

`customer_group`

`customer.create`

`customer.delete`

`customer.info`

`customer.list`

`customer.update`

**Customer Address** ▾

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

## Module: Mage\_Customer

Allows you to export customer groups from Magento

### Resource: customer\_group

#### Method:

- `customer_group.list` (SOAP V1)
- `customerGroupList` (SOAP V2)

Retrieve the list of customer groups

#### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

#### Returns:

| Type  | Name   | Description                     |
|-------|--------|---------------------------------|
| array | result | An array of customerGroupEntity |

The **customerGroupEntity** content is as follows:

| Type   | Name                | Description              |
|--------|---------------------|--------------------------|
| int    | customer_group_id   | ID of the customer group |
| string | customer_group_code | Customer group code      |

## Examples

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'customer_group.list');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->customerGroupList($sessionId);
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->customerGroupList((object)array('sessionId' => $sessionId->result));
var_dump($result->result);
```

## Response Example SOAP V1

```
array
 0 =>
 array
 'customer_group_id' => string '0' (length=1)
 'customer_group_code' => string 'NOT LOGGED IN' (length=13)
 1 =>
 array
 'customer_group_id' => string '1' (length=1)
 'customer_group_code' => string 'General' (length=7)
 2 =>
 array
 'customer_group_id' => string '2' (length=1)
 'customer_group_code' => string 'Wholesale' (length=9)
 3 =>
 array
 'customer_group_id' => string '3' (length=1)
 'customer_group_code' => string 'Retailer' (length=8)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Customer Create

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

customer\_group

**customer.create**

customer.delete

customer.info

customer.list

customer.update

## Customer Address ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Module: Mage\_Customer

Allows you to export/import customers from/to Magento.

## Resource: customer

### Method:

- customer.create (SOAP V1)
- customerCustomerCreate (SOAP V2)

Create a new customer.

### Arguments:

| Type   | Name         | Description                             |
|--------|--------------|-----------------------------------------|
| string | sessionId    | Session ID                              |
| array  | customerData | Array of customerCustomerEntityToCreate |

### Returns:

| Type | Name   | Description                |
|------|--------|----------------------------|
| int  | result | ID of the created customer |

The **customerCustomerEntityToCreate** content is as follows:

| Type   | Name      | Description         |
|--------|-----------|---------------------|
| string | email     | Customer email      |
| string | firstname | Customer first name |

|        |            |                                                  |
|--------|------------|--------------------------------------------------|
| string | lastname   | Customer last name                               |
| string | password   | Customer password                                |
| int    | website_id | Website ID                                       |
| int    | store_id   | Store ID                                         |
| int    | group_id   | Group ID                                         |
| string | prefix     | Customer prefix (optional)                       |
| string | suffix     | Customer suffix (optional)                       |
| string | dob        | Customer date of birth (optional)                |
| string | taxvat     | Customer tax/VAT number (optional)               |
| int    | gender     | Customer gender: 1 - Male, 2 - Female (optional) |
| string | middlename | Customer middle name/initial (optional)          |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->call($session, 'customer.create', array(array('email' =>
'mail@example.org', 'firstname' => 'Dough', 'lastname' => 'Deeks', 'password' =>
'password', 'website_id' => 1, 'store_id' => 1, 'group_id' => 1)));

var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->customerCustomerCreate($session, array('email' => 'customer-
mail@example.org', 'firstname' => 'Dough', 'lastname' => 'Deeks', 'password' =>
'password', 'website_id' => 1, 'store_id' => 1, 'group_id' => 1));

var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));
```

```
$result = $proxy->customerCustomerCreate((object)array('sessionId' => $sessionId->result, 'customerData' => ((object)array('email' => 'customer-mail@example.org', 'firstname' => 'John', 'lastname' => 'Dou', 'password' => '123123', 'website_id' => '0', 'group_id' => '1'))));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Customer Delete

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

customer\_group

customer.create

**customer.delete**

customer.info

customer.list

customer.update

**Customer Address** ▾

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

## Module: Mage\_Customer

Allows you to export/import customers from/to Magento

### Resource: customer

#### Method:

- customer.delete (SOAP V1)
- customerCustomerDelete (SOAP V2)

Delete the required customer.

#### Arguments:

| Type   | Name       | Description |
|--------|------------|-------------|
| string | sessionId  | Session ID  |
| int    | customerId | Customer ID |

#### Returns:

| Type    | Description                     |
|---------|---------------------------------|
| boolean | True if the customer is deleted |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
```

```
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'customer.delete', '2');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->customerCustomerDelete($sessionId, '2');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->customerCustomerDelete((object)array('sessionId' => $sessionId->result, 'customerId' => '2'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Customer Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

customer\_group

customer.create

customer.delete

[customer.info](#)

customer.list

customer.update

### Customer Address ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Customer

Allows you to export/import customers from/to Magento.

### Resource: customer

#### Method:

- customer.info (SOAP V1)
- customerCustomerInfo (SOAP V2)

Retrieve information about the specified customer.

#### Arguments:

| Type          | Name       | Description                 |
|---------------|------------|-----------------------------|
| string        | sessionId  | Session ID                  |
| int           | customerId | ID of the required customer |
| ArrayOfString | attributes | Array of attributes         |

- attributes (optional depending on the version) - only specified attributes will be returned. The customer\_id value is always returned.

#### Returns:

| Type  | Name         | Description                     |
|-------|--------------|---------------------------------|
| array | customerInfo | Array of customerCustomerEntity |

The **customerCustomerEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|         |                     |                                        |
|---------|---------------------|----------------------------------------|
| int     | customer_id         | ID of the customer                     |
| string  | created_at          | Date when the customer was created     |
| string  | updated_at          | Date when the customer was updated     |
| string  | increment_id        | Increment ID                           |
| int     | store_id            | Store ID                               |
| int     | website_id          | Website ID                             |
| string  | created_in          | Store view the customer was created in |
| string  | email               | Customer email                         |
| string  | firstname           | Customer first name                    |
| string  | middlename          | Customer middle name                   |
| string  | lastname            | Customer last name                     |
| int     | group_id            | Customer group ID                      |
| string  | prefix              | Customer prefix                        |
| string  | suffix              | Customer suffix                        |
| string  | dob                 | Customer date of birth                 |
| string  | taxvat              | Tax/VAT number                         |
| boolean | confirmation        | Confirmation flag                      |
| string  | password_hash       | Password hash                          |
| string  | rp_token            | Reset password token                   |
| string  | rp_token_created_at | Date when the password was reset       |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'customer.info', '2');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary
```

```
$result = $proxy->customerCustomerInfo($sessionId, '2');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->customerCustomerInfo((object)array('sessionId' => $sessionId-
>result, 'customerId' => '2'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 'customer_id' => string '2' (length=1)
 'created_at' => string '2012-03-29 12:37:23' (length=19)
 'updated_at' => string '2012-03-30 12:59:21' (length=19)
 'increment_id' => null
 'store_id' => string '2' (length=1)
 'website_id' => string '2' (length=1)
 'confirmation' => null
 'created_in' => string 'English' (length=7)
 'default_billing' => null
 'default_shipping' => string '2' (length=1)
 'disable_auto_group_change' => string '0' (length=1)
 'dob' => null
 'email' => string 'john@example.com' (length=16)
 'firstname' => string 'johny' (length=5)
 'gender' => null
 'group_id' => string '1' (length=1)
 'lastname' => string 'doe' (length=3)
 'middlename' => null
 'password_hash' => string 'cccfb3ecf54c9644a34106783148eff2:sp' (length=35)
 'prefix' => null
 'rp_token' => string '15433dd072f1f4e5aae83231b93f72d0' (length=32)
 'rp_token_created_at' => string '2012-03-30 15:10:31' (length=19)
 'suffix' => null
 'taxvat' => null
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Customer List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

customer\_group

customer.create

customer.delete

customer.info

**customer.list**

customer.update

### Customer Address ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Customer

### Resource: customer

#### Method:

- customer.list (SOAP V1)
- customerCustomerList (SOAP V2)

Allows you to retrieve the list of customers.

#### Arguments:

| Type   | Name      | Description                                        |
|--------|-----------|----------------------------------------------------|
| string | sessionId | Session ID                                         |
| array  | filters   | Array of filters by customer attributes (optional) |

#### Returns:

| Type  | Name      | Description                     |
|-------|-----------|---------------------------------|
| array | storeView | Array of customerCustomerEntity |

The **customerCustomerEntity** content is as follows:

| Type   | Name        | Description                           |
|--------|-------------|---------------------------------------|
| int    | customer_id | ID of the customer                    |
| string | created_at  | Date when the customer was created    |
| string | updated_at  | Date of when the customer was updated |

|         |               |                        |
|---------|---------------|------------------------|
| string  | increment_id  | Increment ID           |
| int     | store_id      | Store ID               |
| int     | website_id    | Website ID             |
| string  | created_in    | Created in             |
| string  | email         | Customer email         |
| string  | firstname     | Customer first name    |
| string  | middlename    | Customer middle name   |
| string  | lastname      | Customer last name     |
| int     | group_id      | Group ID               |
| string  | prefix        | Customer prefix        |
| string  | suffix        | Customer suffix        |
| string  | dob           | Customer date of birth |
| string  | taxvat        | Taxvat value           |
| boolean | confirmation  | Confirmation flag      |
| string  | password_hash | Password hash          |

**Note:** The password\_hash parameter will only match exactly with the same MD5 and salt as was used when Magento stored the value. If you try to match with an unsalted MD5 hash, or any salt other than what Magento used, it will not match. This is just a straight string comparison.

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'customer.list');
var_dump ($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2 (List of All Customers)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->customerCustomerList($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (Complex Filter)

```

$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$complexFilter = array(
 'complex_filter' => array(
 array(
 'key' => 'group_id',
 'value' => array('key' => 'in', 'value' => '1,3')
)
)
);
$result = $client->customerCustomerList($session, $complexFilter);

var_dump ($result);

```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->customerCustomerList((object)array('sessionId' => $sessionId-
>result, 'filters' => null));
var_dump($result->result);

```

### Response Example SOAP V1

```

array
0 =>
array
 'customer_id' => string '2' (length=1)
 'created_at' => string '2012-03-29 12:37:23' (length=19)
 'updated_at' => string '2012-04-03 11:20:18' (length=19)
 'store_id' => string '2' (length=1)
 'website_id' => string '2' (length=1)
 'created_in' => string 'English' (length=7)
 'default_billing' => string '3' (length=1)
 'default_shipping' => string '3' (length=1)
 'disable_auto_group_change' => string '0' (length=1)
 'email' => string 'test@example.com' (length=16)
 'firstname' => string 'John' (length=4)
 'group_id' => string '1' (length=1)
 'lastname' => string 'Doe' (length=3)
 'password_hash' => string 'cccfb3ecf54c9644a34106783148eff2:sp' (length=35)
 'rp_token' => string '15433dd072f1f4e5aae83231b93f72d0' (length=32)
 'rp_token_created_at' => string '2012-03-30 15:10:31' (length=19)

1 =>
array
 'customer_id' => string '4' (length=1)
 'created_at' => string '2012-04-03 11:21:15' (length=19)
 'updated_at' => string '2012-04-03 11:22:57' (length=19)
 'store_id' => string '0' (length=1)
 'website_id' => string '2' (length=1)
 'created_in' => string 'Admin' (length=5)
 'default_billing' => string '8' (length=1)
 'default_shipping' => string '8' (length=1)
 'disable_auto_group_change' => string '0' (length=1)
 'email' => string 'shon@example.com' (length=16)
 'firstname' => string 'Shon' (length=4)

```

```
'group_id' => string '1' (length=1)
'lastname' => string 'McMiland' (length=8)
'password_hash' => string '5670581cabba4e2189e5edee99ed0c86:5q' (length=35)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Customer Update

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

customer\_group

customer.create

customer.delete

customer.info

customer.list

**customer.update**

### Customer Address ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Customer

### Resource: customer

#### Method:

- customer.update (SOAP V1)
- customerCustomerUpdate (SOAP V2)

Update information about the required customer. Note that you need to pass only those arguments which you want to be updated.

#### Arguments:

| Type   | Name         | Description                             |
|--------|--------------|-----------------------------------------|
| string | sessionId    | Session ID                              |
| int    | customerId   | Customer ID                             |
| array  | customerData | Array of customerCustomerEntityToCreate |

#### Returns:

| Type    | Description                     |
|---------|---------------------------------|
| boolean | True if the customer is updated |

The **customerCustomerEntityToCreate** content is as follows:

| Type   | Name        | Description    |
|--------|-------------|----------------|
| int    | customer_id | Customer ID    |
| string | email       | Customer email |

|        |            |                                       |
|--------|------------|---------------------------------------|
| string | firstname  | Customer first name                   |
| string | lastname   | Customer last name                    |
| string | password   | Customer password                     |
| int    | group_id   | Group ID                              |
| string | prefix     | Customer prefix                       |
| string | suffix     | Customer suffix                       |
| string | dob        | Customer date of birth                |
| string | taxvat     | Customer tax/VAT number               |
| int    | gender     | Customer gender: 1 - Male, 2 - Female |
| string | middlename | Customer middle name/initial          |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'customer.update', array('customerId' => '2',
'customerData' => array('firstname' => 'John', 'lastname' => 'Doe', 'email' =>
'test@example.com', 'password' => 'john22')));
var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->customerCustomerUpdate($session, '2', array('email' => 'customer-
mail@example.org', 'firstname' => 'Dough', 'lastname' => 'Deekson', 'password' =>
'password', 'website_id' => 1, 'store_id' => 1, 'group_id' => 1));

var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->customerCustomerUpdate((object)array('sessionId' => $sessionId-
```

```
>result, 'customerId' => '2', 'customerData' => ((object)array(
 'email' => 'customer-mail@example.org',
 'firstname' => 'Dough',
 'lastname' => 'Deekson'
))),
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Customer Address

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

customer\_group

customer.create

customer.delete

customer.info

customer.list

customer.update

## Customer Address ▾

customer\_address.create

customer\_address.delete

customer\_address.info

customer\_address.list

customer\_address.update

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Customer Address

Allows you to create, retrieve, update, and delete address data for a required customer.

**Resource Name:** customer\_address

### Methods:

- [customer\\_address.list](#) - Retrieve the list of customer addresses
- [customer\\_address.create](#) - Create a new address for a customer
- [customer\\_address.info](#) - Retrieve the specified customer address
- [customer\\_address.update](#) - Update the customer address
- [customer\\_address.delete](#) - Delete the customer address

## Faults

| Fault Code | Fault Message                                   |
|------------|-------------------------------------------------|
| 100        | Invalid address data. Details in error message. |
| 101        | Customer not exists.                            |
| 102        | Address not exists.                             |
| 103        | Address not deleted. Details in error message.  |

## Examples

### Example 1. Working with customer address

```
$proxy = new SoapClient('http://magentohost/api/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
```

```

// Create new customer
$newCustomer = array(
 'firstname' => 'First',
 'lastname' => 'Last',
 'email' => 'test@example.com',
 'password' => 'password',
 'store_id' => 0,
 'website_id' => 0
);

$newCustomerId = $proxy->call($sessionId, 'customer.create', array($newCustomer));

//Create new customer address
$newCustomerAddress = array(
 'firstname' => 'First',
 'lastname' => 'Last',
 'country_id' => 'USA',
 'region_id' => '43',
 'region' => 'New York',
 'city' => 'New York',
 'street' => array('street1','street2'),
 'telephone' => '5555-555',
 'postcode' => 10021,

 'is_default_billing' => true,
 'is_default_shipping' => true
);

$newAddressId = $proxy->call($sessionId, 'customer_address.create',
array($newCustomerId, $newCustomerAddress));

var_dump($proxy->call($sessionId, 'customer_address.list', $newCustomerId));

//Update customer address
$proxy->call($sessionId, 'customer_address.update', array($newAddressId,
array('firstname'=>'Changed Firstname')));

var_dump($proxy->call($sessionId, 'customer_address.list', $newCustomerId));

// Delete customer address
$proxy->call($sessionId, 'customer_address.delete', $newAddressId);

var_dump($proxy->call($sessionId, 'customer_address.list', $newCustomerId));

```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Address Create

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

customer\_group

customer.create

customer.delete

customer.info

customer.list

customer.update

### Customer Address ▾

[customer\\_address.create](#)

[customer\\_address.delete](#)

[customer\\_address.info](#)

[customer\\_address.list](#)

[customer\\_address.update](#)

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Customer

### Resource: customer\_address

#### Method:

- `customer_address.create` (SOAP V1)
- `customerAddressCreate` (SOAP V2)

Create a new address for the customer

#### Arguments:

| Type   | Name        | Description                          |
|--------|-------------|--------------------------------------|
| string | sessionId   | Session ID                           |
| int    | customerId  | Customer ID                          |
| array  | addressdata | Array of customerAddressEntityCreate |

#### Returns:

| Type | Name   | Description                        |
|------|--------|------------------------------------|
| int  | result | ID of the created customer address |

The `customerAddressEntityCreate` content is as follows:

| Type   | Name    | Description         |
|--------|---------|---------------------|
| string | city    | Name of the city    |
| string | company | Name of the company |

|               |                     |                                                     |
|---------------|---------------------|-----------------------------------------------------|
| string        | country_id          | Country ID                                          |
| string        | fax                 | Fax                                                 |
| string        | firstname           | Customer first name                                 |
| string        | lastname            | Customer last name                                  |
| string        | middlename          | Customer middle name                                |
| string        | postcode            | Postcode                                            |
| string        | prefix              | Customer prefix                                     |
| int           | region_id           | ID of the region                                    |
| string        | region              | Name of the region                                  |
| ArrayOfString | street              | Array of street addresses                           |
| string        | suffix              | Customer suffix                                     |
| string        | telephone           | Telephone number                                    |
| boolean       | is_default_billing  | True if the address is the default one for billing  |
| boolean       | is_default_shipping | True if the address is the default one for shipping |

**Note:** If you want to leave any address fields empty, specify them as empty ones in the request body.

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call(
$session,
'customer_address.create',
array('customerId' => 2, 'addressdata' => array('firstname' => 'John', 'lastname' =>
'Doe', 'street' => array('Street line 1', 'Street line 2'), 'city' => 'Weaverville',
'country_id' => 'US', 'region' => 'Texas', 'region_id' => 3, 'postcode' => '96093',
'telephone' => '530-623-2513', 'is_default_billing' => FALSE, 'is_default_shipping' =>
FALSE));
var_dump ($result);
```

### Request Example SOAP V2

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->customerAddressCreate($session, '2', array('firstname' => 'John',
'lastname' => 'Doe', 'street' => array('Street line 1', 'Street line 2'), 'city' =>
'Weaverville', 'country_id' => 'US', 'region' => 'Texas', 'region_id' => 3, 'postcode' =>
'96093', 'telephone' => '530-623-2513', 'is_default_billing' => FALSE,
'is_default_shipping' => FALSE));
```

```
var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->customerAddressCreate((object)array('sessionId' => $sessionId-
>result, 'customerId' => '2', 'addressData' => ((object)array(
'city' => 'Weaverville',
'country_id' => 'US',
'postcode' => '96093',
'region' => 'Texas',
'street' => array('Street line 1', 'Streer line 2'),
'telephone' => '847-431-7700',
'lastname' => 'Doe',
'firstname' => 'John',
'is_default_billing' => true
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Address Delete

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

customer\_group

customer.create

customer.delete

customer.info

customer.list

customer.update

### Customer Address ▾

customer\_address.create

**customer\_address.delete**

customer\_address.info

customer\_address.list

customer\_address.update

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Customer

### Resource: customer\_address

#### Method:

- customer\_address.delete (SOAP V1)
- customerAddressDelete (SOAP V2)

Delete the required customer address.

#### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |
| int    | addressId | Address ID  |

#### Returns:

| Type    | Description                             |
|---------|-----------------------------------------|
| boolean | True if the customer address is deleted |

### Examples

#### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
```

```
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'customer_address.delete', '4');
var_dump ($result);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->customerAddressDelete($sessionId, '4');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->customerAddressDelete((object)array('sessionId' => $sessionId-
>result, 'addressId' => '4'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Address Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

customer\_group

customer.create

customer.delete

customer.info

customer.list

customer.update

### Customer Address ▾

customer\_address.create

customer\_address.delete

### customer\_address.info

customer\_address.list

customer\_address.update

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Customer

### Resource: customer\_address

#### Method:

- customer\_address.info (SOAP V1)
- customerAddressInfo (SOAP V2)

Retrieve information about the required customer address.

#### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |
| int    | addressId | Address ID  |

#### Returns:

| Type  | Name | Description                        |
|-------|------|------------------------------------|
| array | info | Array of customerAddressEntityItem |

The **customerAddressEntityItem** content is as follows:

| Type   | Name                | Description                       |
|--------|---------------------|-----------------------------------|
| int    | customer_address_id | ID of the customer address        |
| string | created_at          | Date when the address was created |
| string | updated_at          | Date when the address was updated |

|         |                     |                                                     |
|---------|---------------------|-----------------------------------------------------|
| string  | increment_id        | Increment ID                                        |
| string  | city                | name of the city                                    |
| string  | company             | Name of the company                                 |
| string  | country_id          | ID of the country                                   |
| string  | fax                 | Fax                                                 |
| string  | firstname           | Customer first name                                 |
| string  | lastname            | Customer last name                                  |
| string  | middlename          | Customer middle name                                |
| string  | postcode            | Customer postcode                                   |
| string  | prefix              | Customer prefix                                     |
| string  | region              | Name of the region                                  |
| int     | region_id           | Region ID                                           |
| string  | street              | Name of the street                                  |
| string  | suffix              | Customer suffix                                     |
| string  | telephone           | Telephone number                                    |
| boolean | is_default_billing  | True if the address is the default one for billing  |
| boolean | is_default_shipping | True if the address is the default one for shipping |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'customer_address.info', '2');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->customerAddressInfo($sessionId, '2');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->customerAddressInfo((object)array('sessionId' => $sessionId->result,
'addressId' => '2'));
var_dump($result->result);

```

## Response Example SOAP V1

```

array
 'customer_address_id' => string '2' (length=1)
 'created_at' => string '2012-03-29 13:20:08' (length=19)
 'updated_at' => string '2012-03-29 13:20:08' (length=19)
 'increment_id' => null
 'city' => string 'Las Vegas' (length=9)
 'company' => null
 'country_id' => string 'US' (length=2)
 'fax' => null
 'firstname' => string 'johny' (length=5)
 'lastname' => string 'doe' (length=3)
 'middlename' => null
 'postcode' => string '89032' (length=5)
 'prefix' => null
 'region' => string 'Nevada' (length=6)
 'region_id' => string '39' (length=2)
 'street' => string '3406 Hiney Road' (length=15)
 'suffix' => null
 'telephone' => string '702-283-9556' (length=12)
 'vat_id' => null
 'vat_is_valid' => null
 'vat_request_date' => null
 'vat_request_id' => null
 'vat_request_success' => null
 'is_default_billing' => boolean false
 'is_default_shipping' => boolean true

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Address List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

customer\_group

customer.create

customer.delete

customer.info

customer.list

customer.update

### Customer Address ▾

customer\_address.create

customer\_address.delete

customer\_address.info

**customer\_address.list**

customer\_address.update

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Customer

### Resource: customer\_address

#### Method:

- customer\_address.list (SOAP V1)
- customerAddressList (SOAP V2)

Retrieve the list of customer addresses.

#### Arguments:

| Type   | Name       | Description |
|--------|------------|-------------|
| string | sessionId  | Session ID  |
| int    | customerId | Customer ID |

#### Returns:

| Type  | Name   | Description                    |
|-------|--------|--------------------------------|
| array | result | Array of customerAddressEntity |

The **customerAddressEntity** content is as follows:

| Type   | Name                | Description                       |
|--------|---------------------|-----------------------------------|
| int    | customer_address_id | ID of the customer address        |
| string | created_at          | Date when the address was created |
| string | updated_at          | Date when the address was updated |

|         |                     |                                                     |
|---------|---------------------|-----------------------------------------------------|
| string  | increment_id        | Increment ID                                        |
| string  | city                | City                                                |
| string  | company             | Name of the company                                 |
| string  | country_id          | ID of the country                                   |
| string  | fax                 | Fax                                                 |
| string  | firstname           | Customer first name                                 |
| string  | lastname            | Customer last name                                  |
| string  | middlename          | Customer middle name                                |
| string  | postcode            | Customer postcode                                   |
| string  | prefix              | Customer prefix                                     |
| string  | region              | Name of the region                                  |
| int     | region_id           | Region ID                                           |
| string  | street              | Name of the street                                  |
| string  | suffix              | Customer suffix                                     |
| string  | telephone           | Telephone number                                    |
| boolean | is_default_billing  | True if the address is the default one for billing  |
| boolean | is_default_shipping | True if the address is the default one for shipping |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'customer_address.list', '2');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->customerAddressList($sessionId, '2');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->customerAddressList((object)array('sessionId' => $sessionId->result,
'customerId' => '2'));
var_dump($result->result);

```

## Response Example SOAP V1

```

array
0 =>
array
 'customer_address_id' => string '2' (length=1)
 'created_at' => string '2012-03-29 13:20:08' (length=19)
 'updated_at' => string '2012-03-29 13:39:29' (length=19)
 'city' => string 'Las Vegas' (length=9)
 'country_id' => string 'US' (length=2)
 'firstname' => string 'johny' (length=5)
 'lastname' => string 'doe' (length=3)
 'postcode' => string '89032' (length=5)
 'region' => string 'Nevada' (length=6)
 'region_id' => string '39' (length=2)
 'street' => string '3406 Hiney Road' (length=15)
 'telephone' => string '702-283-9556' (length=12)
 'is_default_billing' => boolean false
 'is_default_shipping' => boolean true
1 =>
array
 'customer_address_id' => string '3' (length=1)
 'created_at' => string '2012-03-29 13:39:29' (length=19)
 'updated_at' => string '2012-03-29 13:39:29' (length=19)
 'city' => string 'Corpus Christi' (length=14)
 'country_id' => string 'US' (length=2)
 'firstname' => string 'johny' (length=5)
 'lastname' => string 'doe' (length=3)
 'postcode' => string '78476' (length=5)
 'region' => string 'Texas' (length=5)
 'region_id' => string '57' (length=2)
 'street' => string '3672 Boone Street' (length=17)
 'telephone' => string '361-280-8488' (length=12)
 'vat_id' => string 'GB999 9999 73' (length=13)
 'is_default_billing' => boolean false
 'is_default_shipping' => boolean false

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Address Update

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

customer\_group

customer.create

customer.delete

customer.info

customer.list

customer.update

### Customer Address ▾

customer\_address.create

customer\_address.delete

customer\_address.info

customer\_address.list

customer\_address.update

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Customer

### Resource: customer\_address

#### Method:

- customer\_address.update (SOAP V1)
- customerAddressUpdate (SOAP V2)

Update address data of the required customer

#### Arguments:

| Type   | Name        | Description                          |
|--------|-------------|--------------------------------------|
| string | sessionId   | Session ID                           |
| int    | addressId   | Address ID                           |
| array  | addressdata | Array of customerAddressEntityCreate |

#### Returns:

| Type    | Description                             |
|---------|-----------------------------------------|
| boolean | True if the customer address is updated |

The **customerAddressEntityCreate** content is as follows:

| Type   | Name    | Description         |
|--------|---------|---------------------|
| string | city    | Name of the city    |
| string | company | Name of the company |

|               |                     |                                                     |
|---------------|---------------------|-----------------------------------------------------|
| string        | country_id          | Country ID                                          |
| string        | fax                 | Fax                                                 |
| string        | firstname           | Customer first name                                 |
| string        | lastname            | Customer last name                                  |
| string        | middlename          | Customer middle name                                |
| string        | postcode            | Postcode                                            |
| string        | prefix              | Customer prefix                                     |
| int           | region_id           | ID of the region                                    |
| string        | region              | Name of the region                                  |
| ArrayOfString | street              | Array of streets                                    |
| string        | suffix              | Customer suffix                                     |
| string        | telephone           | Telephone number                                    |
| boolean       | is_default_billing  | True if the address is the default one for billing  |
| boolean       | is_default_shipping | True if the address is the default one for shipping |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call(
$session,
'customer_address.update',
array('addressId' => 8, 'addressdata' => array('firstname' => 'John', 'lastname' =>
'Doe', 'street' => array('Street line 1', 'Street line 2'), 'city' => 'Weaverville',
'country_id' => 'US', 'region' => 'Texas', 'region_id' => 3, 'postcode' => '96093',
'telephone' => '530-623-2513', 'is_default_billing' => TRUE, 'is_default_shipping' =>
FALSE));
var_dump ($result);
```

### Request Example SOAP V2

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$result = $client->customerAddressUpdate($session, '8', array('firstname' => 'John',
'lastname' => 'Doe', 'street' => array('Street line 1', 'Street line 2'), 'city' =>
'Weaverville', 'country_id' => 'US', 'region' => 'Texas', 'region_id' => 3, 'postcode' =>
'96093', 'telephone' => '530-623-2513', 'is_default_billing' => FALSE,
'is_default_shipping' => FALSE));
```

```
var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->customerAddressUpdate((object)array('sessionId' => $sessionId-
>result, 'addressId' => '8', 'addressData' => ((object)array(
'firstname' => 'John',
'lastname' => 'Doe',
'street' => array('Street line 1', 'Street line 2'),
'city' => 'Weaverville',
'country_id' => 'US',
'region' => 'Texas',
'region_id' => 3,
'postcode' => '96093',
'telephone' => '530-623-2513',
'is_default_billing' => TRUE,
'is_default_shipping' => TRUE
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Directory

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ^

directory\_country.list  
directory\_region.list

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Module: Mage\_Directory

The Mage\_Directory module allows you to retrieve country and region data.

### Country

Allows you to retrieve a list of countries.

**Resource Name:** directory\_country

**Aliases:**

- country

**Methods:**

- [directory\\_country.list](#) - Retrieve a list of countries

### Region

Allows you to retrieve a list of regions within a country.

**Resource Name:** directory\_region

**Aliases:**

- region

**Methods:**

- [directory\\_region.list](#) - Retrieve a list of regions in a specified country

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Country List

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

### Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

## Directory ▾

[directory\\_country.list](#)

[directory\\_region.list](#)

## Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

# Country API

Allows you to retrieve the list of countries from Magento.

## Module: Mage\_Directory

## Resource: directory\_country

### Aliases:

- country

### Method:

- directory\_country.list (SOAP V1)
- directoryCountryList (SOAP V2)

Retrieve the list of countries from Magento.

### Aliases:

- country.list

### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

### Returns:

| Type  | Name      | Description                        |
|-------|-----------|------------------------------------|
| array | countries | An array of directoryCountryEntity |

The **directoryCountryEntity** content is as follows:

| Type   | Name       | Description                   |
|--------|------------|-------------------------------|
| string | country_id | ID of the retrieved country   |
| string | iso2_code  | ISO 2-alpha code              |
| string | iso3_code  | ISO 3-alpha code              |
| string | name       | Name of the retrieved country |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$countries = $proxy->call($sessionId, 'country.list');
var_dump($countries); // Countries list.
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->directoryCountryList($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->directoryCountryList((object)array('sessionId' => $sessionId->result));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'country_id' => string 'AD' (length=2)
 'iso2_code' => string 'AD' (length=2)
 'iso3_code' => string 'AND' (length=3)
 'name' => string 'Andorra' (length=7)
1 =>
array
 'country_id' => string 'AE' (length=2)
 'iso2_code' => string 'AE' (length=2)
 'iso3_code' => string 'ARE' (length=3)
 'name' => string 'United Arab Emirates' (length=20)
2 =>
array
 'country_id' => string 'AF' (length=2)
 'iso2_code' => string 'AF' (length=2)
 'iso3_code' => string 'AFG' (length=3)
```

```
'name' => string 'Afghanistan' (length=11)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Region List

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

## Directory ^

directory\_country.list  
directory\_region.list

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Region API

Allows you to export the list of regions from Magento

## Module: Mage\_Directory

### Resource: directory\_region

#### Aliases:

- region

#### Method:

- directory\_region.list (SOAP V1)
- directoryRegionList (SOAP V2)

Retrieve the list of regions in the specified country.

#### Aliases:

- region.list

#### Arguments:

| Type   | Name      | Description                  |
|--------|-----------|------------------------------|
| string | sessionId | Session ID                   |
| string | country   | Country code in ISO2 or ISO3 |

#### Returns:

| Type  | Name                       | Description                       |
|-------|----------------------------|-----------------------------------|
| array | directoryRegionEntityArray | An array of directoryRegionEntity |

The **directoryRegionEntity** content is as follows:

| Type   | Name      | Description        |
|--------|-----------|--------------------|
| string | region_id | ID of the region   |
| string | code      | Region code        |
| string | name      | Name of the region |

#### Faults:

| Fault Code | Fault Message       |
|------------|---------------------|
| 101        | Country not exists. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$regions = $proxy->call($sessionId, 'region.list', 'US');

var_dump($regions); // Region list for USA.
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->directoryRegionList($sessionId, 'US');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->directoryRegionList((object)array('sessionId' => $sessionId->result,
'country' => 'US'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'region_id' => string '1' (length=1)
 'code' => string 'AL' (length=2)
 'name' => string 'Alabama' (length=7)
1 =>
array
 'region_id' => string '2' (length=1)
 'code' => string 'AK' (length=2)
 'name' => string 'Alaska' (length=6)
```

```
2 =>
 array
 'region_id' => string '3' (length=1)
 'code' => string 'AS' (length=2)
 'name' => string 'American Samoa' (length=14)
3 =>
 array
 'region_id' => string '4' (length=1)
 'code' => string 'AZ' (length=2)
 'name' => string 'Arizona' (length=7)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Sales

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ^

### Sales Order ▾

### Sales Order Credit Memo ▾

### Sales Order Invoice ▾

### Sales Order Shipment ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Module: Mage\_Sales

The Mage\_Sales module allows you to manage sales orders, invoices, shipments, and credit memos.

### Order

Allows you to manage orders.

**Resource Name:** sales\_order

#### Aliases:

- order

#### Methods:

- [sales\\_order.list](#) - Retrieve the list of orders using filters
- [sales\\_order.info](#) - Retrieve the order information
- [sales\\_order.addComment](#) - Add a comment to an order
- [sales\\_order.hold](#) - Hold an order
- [sales\\_order.unhold](#) - Unhold an order
- [sales\\_order.cancel](#) - Cancel an order

### Invoice

Allows you to manage invoices.

**Resource Name:** sales\_order\_invoice

#### Aliases:

- order\_invoice

#### Methods:

- [sales\\_order\\_invoice.list](#) - Retrieve a list of invoices using filters

- [sales\\_order\\_invoice.info](#) - Retrieve information about the invoice
- [sales\\_order\\_invoice.create](#) - Create a new invoice for an order
- [sales\\_order\\_invoice.addComment](#) - Add a new comment to an invoice
- [sales\\_order\\_invoice.capture](#) - Capture an invoice
- [sales\\_order\\_invoice.cancel](#) - Cancel an invoice

## Shipment

Allows you to manage shipments and tracking numbers.

**Resource Name:** sales\_order\_shipment

**Aliases:**

- order\_shipment

**Methods:**

- [sales\\_order\\_shipment.list](#) - Retrieve a list of shipments using filters
- [sales\\_order\\_shipment.info](#) - Retrieve information about the shipment
- [sales\\_order\\_shipment.create](#) - Create a new shipment for an order
- [sales\\_order\\_shipment.addComment](#) - Add a new comment to a shipment
- [sales\\_order\\_shipment.addTrack](#) - Add a new tracking number to a shipment
- [sales\\_order\\_shipment.removeTrack](#) - Remove tracking number from a shipment
- [sales\\_order\\_shipment.getCarriers](#) - Retrieve a list of allowed carriers for an order

## Credit Memo

Allows you to manage order credit memos.

**Resource Name:** sales\_order\_creditmemo

**Aliases:**

- order\_creditmemo

**Methods:**

- [sales\\_order\\_creditmemo.list](#) - Retrieve the list of credit memos by filters
- [sales\\_order\\_creditmemo.info](#) - Retrieve the credit memo information
- [sales\\_order\\_creditmemo.create](#) - Create a new credit memo for order
- [sales\\_order\\_creditmemo.addComment](#) - Add a new comment to the credit memo
- [sales\\_order\\_creditmemo.cancel](#) - Cancel the credit memo

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Sales Order

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

- sales\_order.addComment
- sales\_order.cancel
- sales\_order.hold
- sales\_order.info
- sales\_order.list
- sales\_order.unhold

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

The Mage\_Sales module allows you to manage sales orders, invoices, shipments, and credit memos.

### Order

Allows you to manage orders.

**Resource Name:** sales\_order

#### Aliases:

- order

#### Methods:

- [sales\\_order.list](#) - Retrieve the list of orders using filters
- [sales\\_order.info](#) - Retrieve the order information
- [sales\\_order.addComment](#) - Add a comment to an order
- [sales\\_order.hold](#) - Hold an order
- [sales\\_order.unhold](#) - Unhold an order
- [sales\\_order.cancel](#) - Cancel an order

### Faults

| Fault Code | Fault Message                                       |
|------------|-----------------------------------------------------|
| 100        | Requested order not exists.                         |
| 101        | Invalid filters given. Details in error message.    |
| 102        | Invalid data given. Details in error message.       |
| 103        | Order status not changed. Details in error message. |

## Examples

### Example 1. Work with orders

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

// Getting list of orders created by John Doe
var_dump($proxy->call($sessionId, 'sales_order.list',
array(array('customer_firstname'=>array('eq'=>'John'),
'customer_lastname'=>array('eq'=>'Doe'))));

// Get order info 100000003
var_dump($proxy->call($sessionId, 'sales_order.info', '100000003'));

// Hold order 100000003
$proxy->call($sessionId, 'sales_order.hold', '100000003');

// Unhold order 100000003
$proxy->call($sessionId, 'sales_order.unhold', '100000003');

// Hold order and add comment 100000003
$proxy->call($sessionId, 'sales_order.addComment', array('100000003', 'holded', 'You
order is holded', true));

// Unhold order and add comment 100000003
$proxy->call($sessionId, 'sales_order.addComment', array('100000003', 'pending', 'You
order is pending', true));

// Get order info 100000003
var_dump($proxy->call($sessionId, 'sales_order.info', '100000003'));
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Add Comment

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

- [sales\\_order.addComment](#)
- [sales\\_order.cancel](#)
- [sales\\_order.hold](#)
- [sales\\_order.info](#)
- [sales\\_order.list](#)
- [sales\\_order.unhold](#)

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order

#### Aliases:

- order

#### Method:

- sales\_order.addComment (SOAP V1)
- salesOrderAddComment (SOAP V2)

Allows you to add a new comment to the order.

#### Aliases:

- order.addComment

#### Arguments:

| Type   | Name             | Description                              |
|--------|------------------|------------------------------------------|
| string | sessionId        | Session ID                               |
| string | orderIncrementId | Order increment ID                       |
| string | status           | Order status (pending, processing, etc.) |
| string | comment          | Order comment (optional)                 |
| string | notify           | Notification flag (optional)             |

#### Returns:

| Type | Description |
|------|-------------|
|      |             |

boolean\int

True (1) if the comment is added to the order

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order.addComment', array('orderIncrementId' => '200000004', 'status' => 'processing'));
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderAddComment($sessionId, '200000004', 'processing');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->salesOrderAddComment((object)array('sessionId' => $sessionId->result, 'orderIncrementId' => '200000004', 'status' => 'processing', 'comment' => 'comment to the order', 'notify' => null));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Order Cancel

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

`sales_order.addComment`  
`sales_order.cancel`  
`sales_order.hold`  
`sales_order.info`  
`sales_order.list`  
`sales_order.unhold`

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order

#### Aliases:

- order

#### Method:

- `sales_order.cancel` (SOAP V1)
- `salesOrderCancel` (SOAP V2)

Allows you to cancel the required order.

#### Aliases:

- order.cancel

#### Arguments:

| Type   | Name             | Description        |
|--------|------------------|--------------------|
| string | sessionId        | Session ID         |
| string | orderIncrementId | Order increment ID |

#### Returns:

| Type    | Description                   |
|---------|-------------------------------|
| boolean | True if the order is canceled |

## Examples

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order.cancel', '200000004');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderCancel($sessionId, '200000004');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->salesOrderCancel((object)array('sessionId' => $sessionId->result,
'orderId' => '200000004'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Order Hold

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

`sales_order.addComment`  
`sales_order.cancel`  
`sales_order.hold`  
`sales_order.info`  
`sales_order.list`  
`sales_order.unhold`

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order

### Aliases:

- order

### Method:

- `sales_order.hold` (SOAP V1)
- `salesOrderHold` (SOAP V2)

Allows you to place the required order on hold.

### Aliases:

- order.hold

### Arguments:

| Type   | Name             | Description        |
|--------|------------------|--------------------|
| string | sessionId        | Session ID         |
| string | orderIncrementId | Order increment ID |

### Returns:

| Type        | Description                             |
|-------------|-----------------------------------------|
| boolean\int | True (1) if the order is placed on hold |

## Examples

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order.hold', '200000006');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderHold($sessionId, '200000006');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->salesOrderHold((object)array('sessionId' => $sessionId->result,
'orderIncrementId' => '200000006'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Order Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

- [sales\\_order.addComment](#)
- [sales\\_order.cancel](#)
- [sales\\_order.hold](#)
- [sales\\_order.info](#)
- [sales\\_order.list](#)
- [sales\\_order.unhold](#)

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order

#### Aliases:

- order

#### Method:

- sales\_order.info (SOAP V1)
- salesOrderInfo (SOAP V2)

Allows you to retrieve the required order information.

#### Aliases:

- order.info

#### Arguments:

| Type   | Name             | Description        |
|--------|------------------|--------------------|
| string | sessionId        | Session ID         |
| string | orderIncrementId | Order increment ID |

#### Returns:

| Type  | Name   | Description               |
|-------|--------|---------------------------|
| array | result | Array of salesOrderEntity |

The **salesOrderEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |                             |                                     |
|--------|-----------------------------|-------------------------------------|
| string | increment_id                | Increment ID                        |
| string | parent_id                   | Parent ID                           |
| string | store_id                    | Store ID                            |
| string | created_at                  | Date of creation                    |
| string | updated_at                  | Date of updating                    |
| string | is_active                   | Defines whether the order is active |
| string | customer_id                 | Customer ID                         |
| string | tax_amount                  | Tax amount                          |
| string | shipping_amount             | Shipping amount                     |
| string | discount_amount             | Discount amount                     |
| string | subtotal                    | Subtotal sum                        |
| string | grand_total                 | Grand total sum                     |
| string | total_paid                  | Total paid                          |
| string | total_refunded              | Total refunded                      |
| string | total_qty_ordered           | Total quantity ordered              |
| string | total_canceled              | Total canceled                      |
| string | total_invoiced              | Total invoiced                      |
| string | total_online_refunded       | Total online refunded               |
| string | total_offline_refunded      | Total offline refunded              |
| string | base_tax_amount             | Base tax amount                     |
| string | base_shipping_amount        | Base shipping amount                |
| string | base_discount_amount        | Base discount amount                |
| string | base_subtotal               | Base subtotal                       |
| string | base_grand_total            | Base grand total                    |
| string | base_total_paid             | Base total paid                     |
| string | base_total_refunded         | Base total refunded                 |
| string | base_total_qty_ordered      | Base total quantity ordered         |
| string | base_total_canceled         | Base total canceled                 |
| string | base_total_invoiced         | Base total invoiced                 |
| string | base_total_online_refunded  | Base total online refunded          |
| string | base_total_offline_refunded | Base total offline refunded         |
| string | billing_address_id          | Billing address ID                  |
| string | billing_firstname           | First name in the billing address   |

|        |                      |                                                |
|--------|----------------------|------------------------------------------------|
| string | billing_lastname     | Last name in the billing address               |
| string | shipping_address_id  | Shipping address ID                            |
| string | shipping_firstname   | First name in the shipping address             |
| string | shipping_lastname    | Last name in the shipping address              |
| string | billing_name         | Billing name                                   |
| string | shipping_name        | Shipping name                                  |
| string | store_to_base_rate   | Store to base rate                             |
| string | store_to_order_rate  | Store to order rate                            |
| string | base_to_global_rate  | Base to global rate                            |
| string | base_to_order_rate   | Base to order rate                             |
| string | weight               | Weight                                         |
| string | store_name           | Store name                                     |
| string | remote_ip            | Remote IP                                      |
| string | status               | Order status                                   |
| string | state                | Order state                                    |
| string | applied_rule_ids     | Applied rule IDs                               |
| string | global_currency_code | Global currency code                           |
| string | base_currency_code   | Base currency code                             |
| string | store_currency_code  | Store currency code                            |
| string | order_currency_code  | Order currency code                            |
| string | shipping_method      | Shipping method                                |
| string | shipping_description | Shipping description                           |
| string | customer_email       | Email address of the customer                  |
| string | customer_firstname   | Customer first name                            |
| string | customer_lastname    | Customer last name                             |
| string | quote_id             | Shopping cart ID                               |
| string | is_virtual           | Defines whether the product is a virtual one   |
| string | customer_group_id    | Customer group ID                              |
| string | customer_note_notify | Customer notification                          |
| string | customer_is_guest    | Defines whether the customer is a guest        |
| string | email_sent           | Defines whether the email notification is sent |
| string | order_id             | Order ID                                       |

|        |                  |                                        |
|--------|------------------|----------------------------------------|
| string | gift_message_id  | Gift message ID                        |
| string | gift_message     | Gift message                           |
| array  | shipping_address | Array of salesOrderAddressEntity       |
| array  | billing_address  | Array of salesOrderAddressEntity       |
| array  | items            | Array of salesOrderItemEntity          |
| array  | payment          | Array of salesOrderPaymentEntity       |
| array  | status_history   | Array of salesOrderStatusHistoryEntity |

The **salesOrderAddressEntity** content is as follows:

| Type   | Name         | Description                           |
|--------|--------------|---------------------------------------|
| string | increment_id | Increment ID                          |
| string | parent_id    | Parent ID                             |
| string | created_at   | Date of creation                      |
| string | updated_at   | Date of updating                      |
| string | is_active    | Defines whether the address is active |
| string | address_type | Address type                          |
| string | firstname    | First name                            |
| string | lastname     | Last name                             |
| string | company      | Company name                          |
| string | street       | Street name                           |
| string | city         | City                                  |
| string | region       | Region                                |
| string | postcode     | Post code                             |
| string | country_id   | Country ID                            |
| string | telephone    | Telephone number                      |
| string | fax          | Fax number                            |
| string | region_id    | Region ID                             |
| string | address_id   | Address ID                            |

The **salesOrderItemEntity** content is as follows:

| Type   | Name          | Description           |
|--------|---------------|-----------------------|
| string | item_id       | Item ID               |
| string | order_id      | Order ID              |
| string | quote_item_id | Shopping cart item ID |

|        |                        |                                               |
|--------|------------------------|-----------------------------------------------|
| string | created_at             | Date of creation                              |
| string | updated_at             | Date of updating                              |
| string | product_id             | Product ID                                    |
| string | product_type           | Product type                                  |
| string | product_options        | Product options                               |
| string | weight                 | Weight                                        |
| string | is_virtual             | Defines whether the product is a virtual one  |
| string | sku                    | Product SKU                                   |
| string | name                   | Product name                                  |
| string | applied_rule_ids       | Applied rule IDs                              |
| string | free_shipping          | Defines whether free shipping is applied      |
| string | is_qty_decimal         | Defines whether the items quantity is decimal |
| string | no_discount            | Defines whether no discount is applied        |
| string | qty_canceled           | Items quantity canceled                       |
| string | qty_invoiced           | Items quantity invoiced                       |
| string | qty_ordered            | Items quantity ordered                        |
| string | qty_refunded           | Items quantity refunded                       |
| string | qty_shipped            | Items quantity shipped                        |
| string | cost                   | Cost                                          |
| string | price                  | Price                                         |
| string | base_price             | Base price                                    |
| string | original_price         | Original price                                |
| string | base_original_price    | Base original price                           |
| string | tax_percent            | Tax percent                                   |
| string | tax_amount             | Tax amount                                    |
| string | base_tax_amount        | Base tax amount                               |
| string | tax_invoiced           | Tax invoiced                                  |
| string | base_tax_invoiced      | Base tax invoiced                             |
| string | discount_percent       | Discount percent                              |
| string | discount_amount        | Discount amount                               |
| string | base_discount_amount   | Base discount amount                          |
| string | discount_invoiced      | Discount invoiced                             |
| string | base_discount_invoiced | Base discount invoiced                        |

|        |                                  |                                                         |
|--------|----------------------------------|---------------------------------------------------------|
| string | amount_refunded                  | Amount refunded                                         |
| string | base_amount_refunded             | Base amount refunded                                    |
| string | row_total                        | Row total                                               |
| string | base_row_total                   | Base row total                                          |
| string | row_invoiced                     | Row invoiced                                            |
| string | base_row_invoiced                | Base row invoiced                                       |
| string | row_weight                       | Row weight                                              |
| string | gift_message_id                  | Gift message ID                                         |
| string | gift_message                     | Gift message                                            |
| string | gift_message_available           | Defines whether the gift message is available           |
| string | base_tax_before_discount         | Base tax before discount                                |
| string | tax_before_discount              | Tax before discount                                     |
| string | weee_tax_applied                 | Applied fixed product tax                               |
| string | weee_tax_applied_amount          | Applied fixed product tax amount                        |
| string | weee_tax_applied_row_amount      | Applied fixed product tax row amount                    |
| string | base_weee_tax_applied_amount     | Applied fixed product tax amount (in base currency)     |
| string | base_weee_tax_applied_row_amount | Applied fixed product tax row amount (in base currency) |
| string | weee_tax_disposition             | Fixed product tax disposition                           |
| string | weee_tax_row_disposition         | Fixed product tax row disposition                       |
| string | base_weee_tax_disposition        | Fixed product tax disposition (in base currency)        |
| string | base_weee_tax_row_disposition    | Fixed product tax row disposition (in base currency)    |

The **salesOrderPaymentEntity** content is as follows:

| Type   | Name                 | Description          |
|--------|----------------------|----------------------|
| string | increment_id         | Increment ID         |
| string | parent_id            | Parent ID            |
| string | created_at           | Date of creation     |
| string | updated_at           | Date of updating     |
| string | is_active            | Active flag          |
| string | amount_ordered       | Amount ordered       |
| string | shipping_amount      | Shipping amount      |
| string | base_amount_ordered  | Base amount ordered  |
| string | base_shipping_amount | Base shipping amount |

|        |                   |                                       |
|--------|-------------------|---------------------------------------|
| string | method            | Payment method                        |
| string | po_number         | Purchase order number                 |
| string | cc_type           | Credit card type                      |
| string | cc_number_enc     | Credit card number                    |
| string | cc_last4          | Credit card last 4 digits             |
| string | cc_owner          | Credit card owner                     |
| string | cc_exp_month      | Credit card expiration month          |
| string | cc_exp_year       | Credit card expiration year           |
| string | cc_ss_start_month | Credit card start month (Switch/Solo) |
| string | cc_ss_start_year  | Credit card start year (Switch/Solo)  |
| string | payment_id        | Payment ID                            |

The **salesOrderStatusHistoryEntity** content is as follows:

| Type   | Name                 | Description                              |
|--------|----------------------|------------------------------------------|
| string | increment_id         | Increment ID                             |
| string | parent_id            | Parent ID                                |
| string | created_at           | Date of creation                         |
| string | updated_at           | Date of updating                         |
| string | is_active            | Active flag                              |
| string | is_customer_notified | Defines whether the customer is notified |
| string | status               | Order status                             |
| string | comment              | Order comment                            |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order.info', 'orderIncrementId');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
```

```

$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderInfo($sessionId, '200000006');
var_dump($result);

```

## Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->salesOrderInfo((object)array('sessionId' => $sessionId->result,
'orderIncrementId' => '200000006'));
var_dump($result->result);

```

## Response Example SOAP V1

```

array
 'state' => string 'new' (length=3)
 'status' => string 'pending' (length=7)
 'coupon_code' => null
 'protect_code' => string 'defe18' (length=6)
 'shipping_description' => string 'Flat Rate - Fixed' (length=17)
 'is_virtual' => string '0' (length=1)
 'store_id' => string '2' (length=1)
 'customer_id' => string '2' (length=1)
 'base_discount_amount' => string '0.0000' (length=6)
 'base_discount_canceled' => null
 'base_discount_invoiced' => null
 'base_discount_refunded' => null
 'base_grand_total' => string '619.9700' (length=8)
 'base_shipping_amount' => string '10.0000' (length=7)
 'base_shipping_canceled' => null
 'base_shipping_invoiced' => null
 'base_shipping_refunded' => null
 'base_shipping_tax_amount' => string '0.0000' (length=6)
 'base_shipping_tax_refunded' => null
 'base_subtotal' => string '609.9700' (length=8)
 'base_subtotal_canceled' => null
 'base_subtotal_invoiced' => null
 'base_subtotal_refunded' => null
 'base_tax_amount' => string '0.0000' (length=6)
 'base_tax_canceled' => null
 'base_tax_invoiced' => null
 'base_tax_refunded' => null
 'base_to_global_rate' => string '1.0000' (length=6)
 'base_to_order_rate' => string '1.0000' (length=6)
 'base_total_canceled' => null
 'base_total_invoiced' => null
 'base_total_invoiced_cost' => null
 'base_total_offline_refunded' => null
 'base_total_online_refunded' => null
 'base_total_paid' => null
 'base_total_qty_ordered' => null
 'base_total_refunded' => null
 'discount_amount' => string '0.0000' (length=6)
 'discount_canceled' => null
 'discount_invoiced' => null
 'discount_refunded' => null
 'grand_total' => string '619.9700' (length=8)

```

```
'shipping_amount' => string '10.0000' (length=7)
'shipping_canceled' => null
'shipping_invoiced' => null
'shipping_refunded' => null
'shipping_tax_amount' => string '0.0000' (length=6)
'shipping_tax_refunded' => null
'store_to_base_rate' => string '1.0000' (length=6)
'store_to_order_rate' => string '1.0000' (length=6)
'subtotal' => string '609.9700' (length=8)
'subtotal_canceled' => null
'subtotal_invoiced' => null
'subtotal_refunded' => null
'tax_amount' => string '0.0000' (length=6)
'tax_canceled' => null
'tax_invoiced' => null
'tax_refunded' => null
'total_canceled' => null
'total_invoiced' => null
'total_offline_refunded' => null
'total_online_refunded' => null
'total_paid' => null
'total_qty_ordered' => string '2.0000' (length=6)
'total_refunded' => null
'can_ship_partially' => null
'can_ship_partially_item' => null
'customer_is_guest' => string '0' (length=1)
'customer_note_notify' => string '1' (length=1)
'billing_address_id' => string '1' (length=1)
'customer_group_id' => string '1' (length=1)
'edit_increment' => null
'email_sent' => string '1' (length=1)
'forced_shipment_with_invoice' => null
'payment_auth_expiration' => null
'quote_address_id' => null
'quote_id' => string '1' (length=1)
'shipping_address_id' => string '2' (length=1)
'adjustment_negative' => null
'adjustment_positive' => null
'base_adjustment_negative' => null
'base_adjustment_positive' => null
'base_shipping_discount_amount' => string '0.0000' (length=6)
'base_subtotal_incl_tax' => string '609.9700' (length=8)
'base_total_due' => null
'payment_authorization_amount' => null
'shipping_discount_amount' => string '0.0000' (length=6)
'subtotal_incl_tax' => string '609.9700' (length=8)
'total_due' => null
'weight' => string '1.2000' (length=6)
'customer_dob' => null
'increment_id' => string '200000001' (length=9)
'applied_rule_ids' => null
'base_currency_code' => string 'USD' (length=3)
'customer_email' => string 'john@example.com' (length=16)
'customer_firstname' => string 'johny' (length=5)
'customer_lastname' => string 'doe' (length=3)
'customer_middlename' => null
'customer_prefix' => null
'customer_suffix' => null
'customer_taxvat' => null
'discount_description' => null
'ext_customer_id' => null
'ext_order_id' => null
'global_currency_code' => string 'USD' (length=3)
```

```
'hold_before_state' => null
'hold_before_status' => null
'order_currency_code' => string 'USD' (length=3)
'original_increment_id' => null
'relation_child_id' => null
'relation_child_real_id' => null
'relation_parent_id' => null
'relation_parent_real_id' => null
'remote_ip' => string '127.0.0.1' (length=9)
'shipping_method' => string 'flatrate_flatrate' (length=17)
'store_currency_code' => string 'USD' (length=3)
'store_name' => string 'website

English store
English' (length=29)
 'x_forwarded_for' => null
 'customer_note' => null
 'created_at' => string '2012-03-30 11:40:30' (length=19)
 'updated_at' => string '2012-03-30 11:40:32' (length=19)
 'total_item_count' => string '2' (length=1)
 'customer_gender' => null
 'hidden_tax_amount' => string '0.0000' (length=6)
 'base_hidden_tax_amount' => string '0.0000' (length=6)
 'shipping_hidden_tax_amount' => string '0.0000' (length=6)
 'base_shipping_hidden_tax_amnt' => string '0.0000' (length=6)
 'hidden_tax_invoiced' => null
more elements...
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Order List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

- sales\_order.addComment
- sales\_order.cancel
- sales\_order.hold
- sales\_order.info
- sales\_order.list**
- sales\_order.unhold

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order

#### Aliases:

- order

#### Method:

- sales\_order.list (SOAP V1)
- salesOrderList (SOAP V2)

Allows you to retrieve the list of orders. Additional filters can be applied.

#### Aliases:

- order.list
- salesOrderList (SOAP V2 method name)

#### Arguments:

| Type   | Name      | Description                                              |
|--------|-----------|----------------------------------------------------------|
| string | sessionId | Session ID                                               |
| array  | filters   | Array of filters for the list of sales orders (optional) |

#### Returns:

| Type  | name   | Description               |
|-------|--------|---------------------------|
| array | result | Array of salesOrderEntity |

The **salesOrderEntity** content is as follows:

| Type   | Name                        | Description                         |
|--------|-----------------------------|-------------------------------------|
| string | increment_id                | Increment ID                        |
| string | parent_id                   | Parent ID                           |
| string | store_id                    | Store ID                            |
| string | created_at                  | Date of creation                    |
| string | updated_at                  | Date of updating                    |
| string | is_active                   | Defines whether the order is active |
| string | customer_id                 | Customer ID                         |
| string | tax_amount                  | Tax amount                          |
| string | shipping_amount             | Shipping amount                     |
| string | discount_amount             | Discount amount                     |
| string | subtotal                    | Subtotal sum                        |
| string | grand_total                 | Grand total sum                     |
| string | total_paid                  | Total paid                          |
| string | total_refunded              | Total refunded                      |
| string | total_qty_ordered           | Total quantity ordered              |
| string | total_canceled              | Total canceled                      |
| string | total_invoiced              | Total invoiced                      |
| string | total_online_refunded       | Total online refunded               |
| string | total_offline_refunded      | Total offline refunded              |
| string | base_tax_amount             | Base tax amount                     |
| string | base_shipping_amount        | Base shipping amount                |
| string | base_discount_amount        | Base discount amount                |
| string | base_subtotal               | Base subtotal                       |
| string | base_grand_total            | Base grand total                    |
| string | base_total_paid             | Base total paid                     |
| string | base_total_refunded         | Base total refunded                 |
| string | base_total_qty_ordered      | Base total quantity ordered         |
| string | base_total_canceled         | Base total canceled                 |
| string | base_total_invoiced         | Base total invoiced                 |
| string | base_total_online_refunded  | Base total online refunded          |
| string | base_total_offline_refunded | Base total offline refunded         |
| string | billing_address_id          | Billing address ID                  |

|        |                      |                                                |
|--------|----------------------|------------------------------------------------|
| string | billing_firstname    | First name in the billing address              |
| string | billing_lastname     | Last name in the billing address               |
| string | shipping_address_id  | Shipping address ID                            |
| string | shipping_firstname   | First name in the shipping address             |
| string | shipping_lastname    | Last name in the shipping address              |
| string | billing_name         | Billing name                                   |
| string | shipping_name        | Shipping name                                  |
| string | store_to_base_rate   | Store to base rate                             |
| string | store_to_order_rate  | Store to order rate                            |
| string | base_to_global_rate  | Base to global rate                            |
| string | base_to_order_rate   | Base to order rate                             |
| string | weight               | Weight                                         |
| string | store_name           | Store name                                     |
| string | remote_ip            | Remote IP                                      |
| string | status               | Order status                                   |
| string | state                | Order state                                    |
| string | applied_rule_ids     | Applied rule IDs                               |
| string | global_currency_code | Global currency code                           |
| string | base_currency_code   | Base currency code                             |
| string | store_currency_code  | Store currency code                            |
| string | order_currency_code  | Order currency code                            |
| string | shipping_method      | Shipping method                                |
| string | shipping_description | Shipping description                           |
| string | customer_email       | Email address of the customer                  |
| string | customer_firstname   | Customer first name                            |
| string | customer_lastname    | Customer last name                             |
| string | quote_id             | Shopping cart ID                               |
| string | is_virtual           | Defines whether the product is a virtual one   |
| string | customer_group_id    | Customer group ID                              |
| string | customer_note_notify | Customer notification                          |
| string | customer_is_guest    | Defines whether the customer is a guest        |
| string | email_sent           | Defines whether the email notification is sent |

|        |                 |                 |
|--------|-----------------|-----------------|
| string | order_id        | Order ID        |
| string | gift_message_id | Gift message ID |
| string | gift_message    | Gift message    |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'order.list');
var_dump ($result);
```

### Request Example SOAP V2

```
$params = array(
 'filter' => array(
 array(
 'key' => 'status',
 'value' => 'pending'
),
 array(
 'key' => 'created_at',
 'value' => '2001-11-25 12:12:07',
)
),
 'complex_filter' => array(
 array(
 'key' => 'order_id',
 'value' => array(
 'key' => 'in',
 'value' => '12,23'
),
),
 array(
 'key' => 'protect_code',
 'value' => array(
 'key' => 'eq',
 'value' => 'ebb2a0'
),
),
),
);
);

$result = $client->__call('salesOrderList', $params);
```

### Request Example SOAP V2 (Simple Filter)

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$filter = array('filter' => array(array('key' => 'status', 'value' => 'closed')));
```

```
$result = $client->salesOrderList($session, $filter);

var_dump ($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderList((object)array('sessionId' => $sessionId->result,
'filters' => null));
var_dump($result->result);
```

## XML SOAP V2 Example

### SOAP "v2" XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:Magento"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

<SOAP-ENV:Body>
 <ns1:salesOrderList>
 <sessionId
xsi:type="xsd:string">abbc417256a3ffb93d130a77a2fd3665</sessionId>
 <filters xsi:type="ns1:filters">
 <filter SOAP-ENC:arrayType="ns1:associativeEntity[2]"
xsi:type="ns1:associativeArray">
 <item xsi:type="ns1:associativeEntity">
 <key xsi:type="xsd:string">status</key>
 <value xsi:type="xsd:string">pending</value>
 </item>
 <item xsi:type="ns1:associativeEntity">
 <key xsi:type="xsd:string">created_at</key>
 <value xsi:type="xsd:string">2011-11-29 15:41:11</value>
 </item>
 </filter>
 <complex_filter SOAP-ENC:arrayType="ns1:complexFilter[2]"
xsi:type="ns1:complexFilterArray">
 <item xsi:type="ns1:complexFilter">
 <key xsi:type="xsd:string">order_id</key>
 <value xsi:type="ns1:associativeEntity">
 <key xsi:type="xsd:string">in</key>
 <value xsi:type="xsd:string">Array</value>
 </value>
 </item>
 <item xsi:type="ns1:complexFilter">
 <key xsi:type="xsd:string">protect_code</key>
 <value xsi:type="ns1:associativeEntity">
 <key xsi:type="xsd:string">in</key>
 <value xsi:type="xsd:string">a4ffa8</value>
 </value>
 </item>
 </complex_filter>
 </filters>
 </ns1:salesOrderList>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Order Unhold

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

- [sales\\_order.addComment](#)
- [sales\\_order.cancel](#)
- [sales\\_order.hold](#)
- [sales\\_order.info](#)
- [sales\\_order.list](#)
- [sales\\_order.unhold](#)

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order

#### Aliases:

- order

#### Method:

- sales\_order.unhold (SOAP V1)
- salesOrderUnhold (SOAP V2)

Allows you to unhold the required order.

#### Aliases:

- order.unhold

#### Arguments:

| Type   | Name             | Description        |
|--------|------------------|--------------------|
| string | sessionId        | Session ID         |
| string | orderIncrementId | Order increment ID |

#### Returns:

| Type        | Description                     |
|-------------|---------------------------------|
| boolean\int | True (1) if the order is unheld |

## Examples

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order.unhold', '200000006');
var_dump($result);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->salesOrderUnhold($sessionId, '200000006');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderUnhold((object)array('sessionId' => $sessionId->result,
'orderIncrementId' => '200000006'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Sales Order Credit Memo

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▲

[Sales Order](#) ▾

[Sales Order Credit Memo](#) ▾

`sales_order_creditmemo.addC...`  
`sales_order_creditmemo.cancel`  
`sales_order_creditmemo.create`  
`sales_order_creditmemo.info`  
`sales_order_creditmemo.list`

[Sales Order Invoice](#) ▾

[Sales Order Shipment](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

### Module: Sales\_Order\_Creditmemo

Allows you to operate with credit memos for orders.

Resource: sales\_order\_creditmemo

**Aliases:** order\_creditmemo

#### Methods:

- [sales\\_order\\_creditmemo.list](#) - Retrieve the list of credit memos by filters
- [sales\\_order\\_creditmemo.info](#) - Retrieve the credit memo information
- [sales\\_order\\_creditmemo.create](#) - Create a new credit memo for order
- [sales\\_order\\_creditmemo.addComment](#) - Add a new comment to the credit memo
- [sales\\_order\\_creditmemo.cancel](#) - Cancel the credit memo

#### Faults:

| Fault Code | Fault Message                                                                        |
|------------|--------------------------------------------------------------------------------------|
| 100        | Requested credit memo does not exist.                                                |
| 101        | Invalid filter given. Details in error message.                                      |
| 102        | Invalid data given. Details in error message.                                        |
| 103        | Requested order does not exist.                                                      |
| 104        | Credit memo status not changed.                                                      |
| 105        | Money can not be refunded to the store credit account as order was created by guest. |
| 106        | Credit memo for requested order can not be created.                                  |

### Example:

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

// Create creditmemo
$orderIncrementId = '100000683'; //increment id of the invoiced order
$data = array(
 'qty' => array(
 '712' => 1
),
 'shipping_amount' => 3,
 'adjustment_positive' => 0.7,
 'adjustment_negative' => 0.06
);
$creditmemoIncrementId = $proxy->call($sessionId, 'order_creditmemo.create',
array($orderIncrementId, $data));
echo $creditmemoIncrementId . "
";

// Add comment to created creditmemo
$commentText = "Credit memo comment successfully added";
$isCommentAdded = $proxy->call($sessionId, 'order_creditmemo.addComment',
array($creditmemoIncrementId, $commentText, true));

// Retrieve information about created creditmemo
$creditmemoInfo = $proxy->call($sessionId, 'order_creditmemo.info',
array($creditmemoIncrementId));
print_r($creditmemoInfo);

// Retrieve list of creditmemos by filter
$filter = array(
 'increment_id' => array(
 'or' => array(
 array(
 'from' => '100000617',
 'to' => '100000619',
),
 array(
 'from' => $creditmemoIncrementId,
 'to' => NULL,
)
)
)
);
$creditmemoList = $proxy->call($sessionId, 'order_creditmemo.list', array($filter));
print_r($creditmemoList);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Add Comment

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

##### Sales Order Credit Memo ▾

[sales\\_order\\_creditmemo.addC...](#)  
sales\_order\_creditmemo.cancel  
sales\_order\_creditmemo.create  
sales\_order\_creditmemo.info  
sales\_order\_creditmemo.list

##### Sales Order Invoice ▾

##### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Order Credit Memo API

Resource: sales\_order\_creditmemo

**Aliases:** order\_creditmemo

### Method:

- order\_creditmemo.addComment (SOAP V1)
- salesOrderCreditmemoAddComment (SOAP V2)

Allows you to add a new comment to an existing credit memo. Email notification can be sent to the user email.

### Arguments:

| Type   | Name                  | Description                                                 |
|--------|-----------------------|-------------------------------------------------------------|
| string | sessionId             | Session ID                                                  |
| string | creditmemoincrementId | Credit memo increment ID                                    |
| string | comment               | Comment text (optional)                                     |
| int    | notifyCustomer        | Notify customer by email flag (optional)                    |
| int    | includeComment        | Include comment text into the email notification (optional) |

### Return:

| Type        | Description                                         |
|-------------|-----------------------------------------------------|
| boolean\int | True (1) if the comment is added to the credit memo |

### Faults:

| Fault Code | Fault Message |
|------------|---------------|
|            |               |

|     |                                               |
|-----|-----------------------------------------------|
| 100 | Requested credit memo does not exist.         |
| 102 | Invalid data given. Details in error message. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$creditmemoIncrementId = '200000001'; //increment id of existing credit memo
$commentText = "Credit memo comment successfully added";

$isCommentAdded = $proxy->call($sessionId, 'order_creditmemo.addComment',
array($creditmemoIncrementId, $commentText, true));
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderCreditmemoAddComment($sessionId, '200000001');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderCreditmemoAddComment((object)array('sessionId' =>
$sessionId->result, 'creditmemoIncrementId' => '200000001', 'comment' => 'credit memo
comment', 'notifyCustomer' => 1, 'includeComment' => 1));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Memo Cancel

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▲

**Sales Order** ▾

**Sales Order Credit Memo** ▾

[sales\\_order\\_creditmemo.addC...](#)  
[sales\\_order\\_creditmemo.cancel](#)  
[sales\\_order\\_creditmemo.create](#)  
[sales\\_order\\_creditmemo.info](#)  
[sales\\_order\\_creditmemo.list](#)

**Sales Order Invoice** ▾

**Sales Order Shipment** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

**Module:** Order Credit Memo API

**Resource:** sales\_order\_creditmemo

**Aliases:** order\_creditmemo

**Method:**

- order\_creditmemo.cancel (SOAP V1)
- salesOrderCreditmemoCancel (SOAP V2)

Allows you to cancel an existing credit memo.

**Arguments:**

| Type   | Name                  | Description              |
|--------|-----------------------|--------------------------|
| string | sessionId             | Session ID               |
| string | creditmemoincrementId | Credit memo increment ID |

**Return:**

| Type   | Name   | Description                         |
|--------|--------|-------------------------------------|
| string | result | Result of canceling the credit memo |

**Faults:**

| Fault Code | Fault Message                         |
|------------|---------------------------------------|
| 100        | Requested credit memo does not exist. |
| 104        | Credit memo status not changed.       |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$creditmemoIncrementId = '100000637'; //increment id of existing credit memo

$isCreditMemoCanceled = $proxy->call($sessionId, 'order_creditmemo.cancel',
array($creditmemoIncrementId));
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Memo Create

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▾

**Sales Order** ▾

**Sales Order Credit Memo** ▾

sales\_order\_creditmemo.addC...  
sales\_order\_creditmemo.cancel  
**sales\_order\_creditmemo.create**  
sales\_order\_creditmemo.info  
sales\_order\_creditmemo.list

**Sales Order Invoice** ▾

**Sales Order Shipment** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

**Module:** Order Credit Memo API

**Resource:** sales\_order\_creditmemo

**Aliases:** `order_creditmemo`

**Method:**

- `order_creditmemo.create` (SOAP V1)
- `salesOrderCreditmemoCreate` (SOAP V2)

Allows you to create a new credit memo for the invoiced order. Comments can be added and an email notification can be sent to the user email.

**Arguments:**

| Type   | Name                      | Description                                                           |
|--------|---------------------------|-----------------------------------------------------------------------|
| string | sessionId                 | Session ID                                                            |
| string | orderIncrementId          | Order increment ID                                                    |
| array  | creditmemoData            | Array of salesOrderCreditmemoData (optional)                          |
| string | comment                   | Comment text (optional)                                               |
| int    | notifyCustomer            | Notify customer by email flag (optional)                              |
| int    | includeComment            | Include comment text into an email notification (optional)            |
| string | refundToStoreCreditAmount | Payment amount to be refunded to the customer store credit (optional) |

**Return:**

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |        |                                  |
|--------|--------|----------------------------------|
| string | result | Created credit memo increment ID |
|--------|--------|----------------------------------|

The **salesOrderCreditmemoData** content is as follows:

| Type   | Name                | Description                         |
|--------|---------------------|-------------------------------------|
| array  | qtys                | Array of orderItemIdQty             |
| double | shipping_amount     | Refund shipping amount (optional)   |
| double | adjustment_positive | Adjustment refund amount (optional) |
| double | adjustment_negative | Adjustment fee amount (optional)    |

The **orderItemIdQty** content is as follows:

| Type   | Name          | Description                   |
|--------|---------------|-------------------------------|
| int    | order_item_id | Order item ID to be refunded  |
| double | qty           | Items quantity to be refunded |

### Faults:

| Fault Code | Fault Message                                                                        |
|------------|--------------------------------------------------------------------------------------|
| 102        | Invalid data given. Details in error message.                                        |
| 103        | Requested order does not exist.                                                      |
| 105        | Money can not be refunded to the store credit account as order was created by guest. |
| 106        | Credit memo for requested order can not be created.                                  |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'order_creditmemo.create', '200000010');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderCreditmemoCreate($sessionId, '200000010');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
```

```
$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->salesOrderCreditmemoCreate((object)array('sessionId' => $sessionId->result, 'creditmemoIncrementId' => '200000010',
 'creditmemoData' => array(
 'qtys' => array(
 'order_item_id' => 3,
 'qty' => '1'),
 'shipping_amount' => null,
 'adjustment_positive' => '0',
 'adjustment_negative' => null),
 'comment' => 'comment for credit memo',
 'notifyCustomer' => null,
 'includeComment' => 1,
 'refundToStoreCreditAmount' => '1'
)));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Memo Info

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▾

**Sales Order** ▾

**Sales Order Credit Memo** ▾

`sales_order_creditmemo.addC...`  
`sales_order_creditmemo.cancel`  
`sales_order_creditmemo.create`  
`sales\_order\_creditmemo.info`  
`sales_order_creditmemo.list`

**Sales Order Invoice** ▾

**Sales Order Shipment** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

**Module:** Order Credit Memo API

**Resource:** sales\_order\_creditmemo

**Aliases:** `order_creditmemo`

**Method:**

- `order_creditmemo.info` (SOAP V1)
- `salesOrderCreditmemoInfo` (SOAP V2)

Allows you to retrieve full information about the specified credit memo.

**Arguments:**

| Type   | Name                  | Description              |
|--------|-----------------------|--------------------------|
| string | sessionId             | Session ID               |
| string | creditmemoIncrementId | Credit memo increment ID |

**Return:**

| Type  | Name   | Description                         |
|-------|--------|-------------------------------------|
| array | result | Array of salesOrderCreditmemoEntity |

The `salesOrderCreditmemoEntity` content is as follows:

| Type   | Name       | Description      |
|--------|------------|------------------|
| string | updated_at | Date of updating |
| string | created_at | Date of creation |

|        |                          |                                                         |
|--------|--------------------------|---------------------------------------------------------|
| string | increment_id             | Increment ID                                            |
| string | transaction_id           | Transaction ID                                          |
| string | global_currency_code     | Global currency code                                    |
| string | base_currency_code       | Base currency code                                      |
| string | order_currency_code      | Order currency code                                     |
| string | store_currency_code      | Store currency code                                     |
| string | cybersource_token        | Cybersource token                                       |
| string | invoice_id               | ID of the invoice for which the credit memo was created |
| string | billing_address_id       | Billing address ID                                      |
| string | shipping_address_id      | Shipping address ID                                     |
| string | state                    | State                                                   |
| string | creditmemo_status        | Credit memo status                                      |
| string | email_sent               | Defines whether the email is sent                       |
| string | order_id                 | ID of the order for which the credit memo was created   |
| string | tax_amount               | Tax amount                                              |
| string | shipping_tax_amount      | Shipping tax amount                                     |
| string | base_tax_amount          | Base tax amount                                         |
| string | base_adjustment_positive | Adjustment refund amount (using base currency)          |
| string | base_grand_total         | Base grand total                                        |
| string | adjustment               | Adjustment                                              |
| string | subtotal                 | Subtotal                                                |
| string | discount_amount          | Discount amount                                         |
| string | base_subtotal            | Base subtotal                                           |
| string | base_adjustment          | Base adjustment                                         |
| string | base_to_global_rate      | Base to global rate                                     |
| string | store_to_base_rate       | Store to base rate                                      |
| string | base_shipping_amount     | Base shipping amount                                    |
| string | adjustment_negative      | Adjustment fee amount                                   |
| string | subtotal_incl_tax        | Subtotal including tax                                  |
| string | shipping_amount          | Shipping amount                                         |
| string | base_subtotal_incl_tax   | Base subtotal including tax                             |
| string | base_adjustment_negative | Adjustment fee amount (using base currency)             |
| string | grand_total              | Grand total                                             |

|        |                                |                                                           |
|--------|--------------------------------|-----------------------------------------------------------|
| string | base_discount_amount           | Base discount amount                                      |
| string | base_to_order_rate             | Base to order rate                                        |
| string | store_to_order_rate            | Store to order rate                                       |
| string | base_shipping_tax_amount       | Base shipping tax amount                                  |
| string | adjustment_positive            | Adjustment refund amount                                  |
| string | store_id                       | Store ID                                                  |
| string | hidden_tax_amount              | Hidden tax amount                                         |
| string | base_hidden_tax_amount         | Base hidden tax amount                                    |
| string | shipping_hidden_tax_amount     | Shipping hidden tax amount                                |
| string | base_shipping_hidden_tax_amnt  | Base shipping hidden tax amount                           |
| string | shipping_incl_tax              | Shipping including tax                                    |
| string | base_shipping_incl_tax         | Base shipping including tax                               |
| string | base_customer_balance_amount   | Base customer balance amount                              |
| string | customer_balance_amount        | Customer balance amount                                   |
| string | bs_customer_bal_total_refunded | Refunded base customer balance amount                     |
| string | customer_bal_total_refunded    | Customer balance total refunded                           |
| string | base_gift_cards_amount         | Base gift cards amount                                    |
| string | gift_cards_amount              | Gift cards amount                                         |
| string | gw_base_price                  | Gift wrapping price refunded amount (using base currency) |
| string | gw_price                       | Gift wrapping price refunded amount                       |
| string | gw_items_base_price            | Gift wrapping items base price                            |
| string | gw_items_price                 | Gift wrapping items price                                 |
| string | gw_card_base_price             | Gift wrapping card base price                             |
| string | gw_card_price                  | Gift wrapping card price                                  |
| string | gw_base_tax_amount             | Gift wrapping tax amount refunded (using base currency)   |
| string | gw_tax_amount                  | Gift wrapping tax amount refunded                         |
| string | gw_items_base_tax_amount       | Gift wrapping items base tax amount                       |
| string | gw_items_tax_amount            | Gift wrapping items tax amount                            |
| string | gw_card_base_tax_amount        | Gift wrapping card base tax amount                        |
| string | gw_card_tax_amount             | Gift wrapping card tax amount                             |
| string | base_reward_currency_amount    | Base reward currency amount                               |
| string | reward_currency_amount         | Reward currency amount                                    |
|        |                                |                                                           |

|        |                              |                                            |
|--------|------------------------------|--------------------------------------------|
| string | reward_points_balance        | Reward points balance                      |
| string | reward_points_balance_refund | Reward points balance refund               |
| string | creditmemo_id                | Credit memo ID                             |
| array  | items                        | Array of salesOrderCreditmemoItemEntity    |
| array  | comments                     | Array of salesOrderCreditmemoCommentEntity |

The **salesOrderCreditmemoItemEntity** content is as follows:

| Type   | Name                              | Description                                             |
|--------|-----------------------------------|---------------------------------------------------------|
| string | item_id                           | Credit memo item ID                                     |
| string | parent_id                         | Parent ID                                               |
| string | weeee_tax_applied_row_amount      | Applied fixed product tax row amount                    |
| string | base_price                        | Base price                                              |
| string | base_weeee_tax_row_disposition    | Fixed product tax row disposition (in base currency)    |
| string | tax_amount                        | Tax amount                                              |
| string | base_weeee_tax_applied_amount     | Applied fixed product tax amount (in base currency)     |
| string | weeee_tax_row_disposition         | Fixed product tax row disposition                       |
| string | base_row_total                    | Base row total                                          |
| string | discount_amount                   | Discount amount                                         |
| string | row_total                         | Row total                                               |
| string | weeee_tax_applied_amount          | Applied fixed product tax amount                        |
| string | base_discount_amount              | Base discount amount                                    |
| string | base_weeee_tax_disposition        | Fixed product tax disposition (in base currency)        |
| string | price_incl_tax                    | Price including tax                                     |
| string | base_tax_amount                   | Base tax amount                                         |
| string | weeee_tax_disposition             | Fixed product tax disposition                           |
| string | base_price_incl_tax               | Base price including tax                                |
| string | qty                               | Quantity                                                |
| string | base_cost                         | Base cost                                               |
| string | base_weeee_tax_applied_row_amount | Applied fixed product tax row amount (in base currency) |
| string | price                             | Price                                                   |
| string | base_row_total_incl_tax           | Base row total including tax                            |
| string | row_total_incl_tax                | Row total including tax                                 |
| string | product_id                        | Product ID                                              |
| string | order_item_id                     | Order item ID                                           |

|        |                        |                           |
|--------|------------------------|---------------------------|
| string | additional_data        | Additional data           |
| string | description            | Description               |
| string | weee_tax_applied       | Applied fixed product tax |
| string | sku                    | Item SKU                  |
| string | name                   | Name                      |
| string | hidden_tax_amount      | Hidden tax amount         |
| string | base_hidden_tax_amount | Base hidden tax amount    |

The **salesOrderCreditmemoCommentEntity** content is as follows:

| Type   | Name                 | Description                                            |
|--------|----------------------|--------------------------------------------------------|
| string | parent_id            | Parent ID                                              |
| string | created_at           | Date of creation                                       |
| string | comment              | Comment data                                           |
| string | is_customer_notified | Defines whether the customer is notified               |
| string | comment_id           | Comment ID                                             |
| string | is_visible_on_front  | Defines whether the comment is visible on the frontend |

#### Faults:

| Fault Code | Fault Description                     |
|------------|---------------------------------------|
| 100        | Requested credit memo does not exist. |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'order_creditmemo.info', '200000001');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderCreditmemoInfo($sessionId, '200000001');
var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->salesOrderCreditmemoInfo((object)array('sessionId' => $sessionId->result, 'creditmemoIncrementId' => '200000001'));
var_dump($result->result);

```

## Response Example SOAP V1

```

array(
 'store_id' => '1',
 'adjustment_positive' => NULL,
 'base_shipping_tax_amount' => '0.0000',
 'store_to_order_rate' => '1.0000',
 'base_discount_amount' => '0.0000',
 'base_to_order_rate' => '1.0000',
 'grand_total' => '90.0000',
 'base_adjustment_negative' => NULL,
 'base_subtotal_incl_tax' => '75.0000',
 'shipping_amount' => '15.0000',
 'subtotal_incl_tax' => '75.0000',
 'adjustment_negative' => NULL,
 'base_shipping_amount' => '15.0000',
 'store_to_base_rate' => '1.0000',
 'base_to_global_rate' => '1.0000',
 'base_adjustment' => '0.0000',
 'base_subtotal' => '75.0000',
 'discount_amount' => '0.0000',
 'subtotal' => '75.0000',
 'adjustment' => '0.0000',
 'base_grand_total' => '90.0000',
 'base_adjustment_positive' => NULL,
 'base_tax_amount' => '0.0000',
 'shipping_tax_amount' => '0.0000',
 'tax_amount' => '0.0000',
 'order_id' => '744',
 'email_sent' => NULL,
 'creditmemo_status' => NULL,
 'state' => '2',
 'shipping_address_id' => '1488',
 'billing_address_id' => '1487',
 'invoice_id' => NULL,
 'cybersource_token' => NULL,
 'store_currency_code' => 'USD',
 'order_currency_code' => 'USD',
 'base_currency_code' => 'USD',
 'global_currency_code' => 'USD',
 'transaction_id' => NULL,
 'increment_id' => '100000684',
 'created_at' => '2011-05-27 10:53:03',
 'updated_at' => '2011-05-27 10:53:03',
 'hidden_tax_amount' => '0.0000',
 'base_hidden_tax_amount' => '0.0000',
 'shipping_hidden_tax_amount' => NULL,
 'base_shipping_hidden_tax_amnt' => NULL,
 'shipping_incl_tax' => '15.0000',
 'base_shipping_incl_tax' => '15.0000',
 'base_customer_balance_amount' => NULL,
 'customer_balance_amount' => NULL,
 'bs_customer_bal_total_refunded' => '0.0000',
)

```

```

'customer_bal_total_refunded' => '0.0000',
'base_gift_cards_amount' => NULL,
'gift_cards_amount' => NULL,
'gw_base_price' => NULL,
'gw_price' => NULL,
'gw_items_base_price' => NULL,
'gw_items_price' => NULL,
'gw_card_base_price' => NULL,
'gw_card_price' => NULL,
'gw_base_tax_amount' => NULL,
'gw_tax_amount' => NULL,
'gw_items_base_tax_amount' => NULL,
'gw_items_tax_amount' => NULL,
'gw_card_base_tax_amount' => NULL,
'gw_card_tax_amount' => NULL,
'base_reward_currency_amount' => NULL,
'reward_currency_amount' => NULL,
'reward_points_balance' => NULL,
'reward_points_balance_refund' => NULL,
'base_customer_balance_total_refunded' => '0.0000',
'customer_balance_total_refunded' => '0.0000',
'gw_printed_card_base_price' => NULL,
'gw_printed_card_price' => NULL,
'gw_printed_card_base_tax_amount' => NULL,
'gw_printed_card_tax_amount' => NULL,
'reward_points_balance_to_refund' => NULL,
'creditmemo_id' => '684',
'order_increment_id' => '100000744',
'items' => array(
 0 => array(
 'parent_id' => '684',
 'weeee_tax_applied_row_amount' => '0.0000',
 'base_price' => '55.0000',
 'base_weeee_tax_row_disposition' => '0.0000',
 'tax_amount' => '0.0000',
 'base_weeee_tax_applied_amount' => '0.0000',
 'weeee_tax_row_disposition' => '0.0000',
 'base_row_total' => '55.0000',
 'discount_amount' => NULL,
 'row_total' => '55.0000',
 'weeee_tax_applied_amount' => '0.0000',
 'base_discount_amount' => NULL,
 'base_weeee_tax_disposition' => '0.0000',
 'price_incl_tax' => '55.0000',
 'base_tax_amount' => '0.0000',
 'weeee_tax_disposition' => '0.0000',
 'base_price_incl_tax' => '55.0000',
 'qty' => '1.0000',
 'base_cost' => NULL,
 'base_weeee_tax_applied_row_amnt' => NULL,
 'price' => '55.0000',
 'base_row_total_incl_tax' => '55.0000',
 'row_total_incl_tax' => '55.0000',
 'product_id' => '20',
 'order_item_id' => '775',
 'additional_data' => NULL,
 'description' => NULL,
 'weeee_tax_applied' => 'a:0:{}',
 'sku' => 'test_product_sku',
 'name' => 'Test Product',
 'hidden_tax_amount' => '0.0000',
 'base_hidden_tax_amount' => '0.0000',
 'item_id' => '708'
)
)

```

```

),
1 => array(
 'parent_id' => '684',
 'weeee_tax_applied_row_amount' => '0.0000',
 'base_price' => '10.0000',
 'base_weeee_tax_row_disposition' => '0.0000',
 'tax_amount' => '0.0000',
 'base_weeee_tax_applied_amount' => '0.0000',
 'weeee_tax_row_disposition' => '0.0000',
 'base_row_total' => '20.0000',
 'discount_amount' => NULL,
 'row_total' => '20.0000',
 'weeee_tax_applied_amount' => '0.0000',
 'base_discount_amount' => NULL,
 'base_weeee_tax_disposition' => '0.0000',
 'price_incl_tax' => '10.0000',
 'base_tax_amount' => '0.0000',
 'weeee_tax_disposition' => '0.0000',
 'base_price_incl_tax' => '10.0000',
 'qty' => '2.0000',
 'base_cost' => NULL,
 'base_weeee_tax_applied_row_amnt' => NULL,
 'price' => '10.0000',
 'base_row_total_incl_tax' => '20.0000',
 'row_total_incl_tax' => '20.0000',
 'product_id' => '21',
 'order_item_id' => '776',
 'additional_data' => NULL,
 'description' => NULL,
 'weeee_tax_applied' => 'a:0:{}',
 'sku' => 'test_product_sku_2',
 'name' => 'Test 2',
 'hidden_tax_amount' => '0.0000',
 'base_hidden_tax_amount' => '0.0000',
 'item_id' => '709'
)
),
'comments' => array(
 0 => array(
 'parent_id' => '684',
 'is_customer_notified' => '0',
 'is_visible_on_front' => '0',
 'comment' => 'Test CreditMemo successfully created',
 'created_at' => '2011-05-27 10:53:03',
 'comment_id' => '118'
),
 1 => array(
 'parent_id' => '684',
 'is_customer_notified' => '0',
 'is_visible_on_front' => '0',
 'comment' => 'Test CreditMemo comment successfully added',
 'created_at' => '2011-05-27 10:53:03',
 'comment_id' => '119'
)
)
);

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Memo List

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▲

[Sales Order](#) ▾

[Sales Order Credit Memo](#) ▾

[sales\\_order\\_creditmemo.addC...](#)  
[sales\\_order\\_creditmemo.cancel](#)  
[sales\\_order\\_creditmemo.create](#)  
[sales\\_order\\_creditmemo.info](#)  
[sales\\_order\\_creditmemo.list](#)

[Sales Order Invoice](#) ▾

[Sales Order Shipment](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

**Module:** Order Credit Memo API

**Resource:** sales\_order\_creditmemo

**Aliases:** order\_creditmemo

**Method:**

- order\_creditmemo.list (SOAP V1)
- salesOrderCreditmemoList (SOAP V2)

Allows you to retrieve the list of credit memos by filters.

**Arguments:**

| Type              | Name      | Description        |
|-------------------|-----------|--------------------|
| string            | sessionId | Session ID         |
| associative array | filters   | Filters (optional) |

**Return:**

| Type  | Name   | Description                         |
|-------|--------|-------------------------------------|
| array | result | Array of salesOrderCreditmemoEntity |

The **salesOrderCreditmemoEntity** content is as follows:

| Type   | Name       | Description      |
|--------|------------|------------------|
| string | updated_at | Date of updating |
| string | created_at | Date of creation |

|        |                          |                                                         |
|--------|--------------------------|---------------------------------------------------------|
| string | increment_id             | Increment ID                                            |
| string | transaction_id           | Transaction ID                                          |
| string | global_currency_code     | Global currency code                                    |
| string | base_currency_code       | Base currency code                                      |
| string | order_currency_code      | Order currency code                                     |
| string | store_currency_code      | Store currency code                                     |
| string | cybersource_token        | Cybersource token                                       |
| string | invoice_id               | ID of the invoice for which the credit memo was created |
| string | billing_address_id       | Billing address ID                                      |
| string | shipping_address_id      | Shipping address ID                                     |
| string | state                    | State                                                   |
| string | creditmemo_status        | Credit memo status                                      |
| string | email_sent               | Defines whether the email is sent                       |
| string | order_id                 | ID of the order for which the credit memo was created   |
| string | tax_amount               | Tax amount                                              |
| string | shipping_tax_amount      | Shipping tax amount                                     |
| string | base_tax_amount          | Base tax amount                                         |
| string | base_adjustment_positive | Adjustment refund amount (using base currency)          |
| string | base_grand_total         | Base grand total                                        |
| string | adjustment               | Adjustment                                              |
| string | subtotal                 | Subtotal                                                |
| string | discount_amount          | Discount amount                                         |
| string | base_subtotal            | Base subtotal                                           |
| string | base_adjustment          | Base adjustment                                         |
| string | base_to_global_rate      | Base to global rate                                     |
| string | store_to_base_rate       | Store to base rate                                      |
| string | base_shipping_amount     | Base shipping amount                                    |
| string | adjustment_negative      | Adjustment fee amount                                   |
| string | subtotal_incl_tax        | Subtotal including tax                                  |
| string | shipping_amount          | Shipping amount                                         |
| string | base_subtotal_incl_tax   | Base subtotal including tax                             |
| string | base_adjustment_negative | Adjustment fee amount (using base currency)             |
| string | grand_total              | Grand total                                             |

|        |                                |                                                           |
|--------|--------------------------------|-----------------------------------------------------------|
| string | base_discount_amount           | Base discount amount                                      |
| string | base_to_order_rate             | Base to order rate                                        |
| string | store_to_order_rate            | Store to order rate                                       |
| string | base_shipping_tax_amount       | Base shipping tax amount                                  |
| string | adjustment_positive            | Adjustment refund amount                                  |
| string | store_id                       | Store ID                                                  |
| string | hidden_tax_amount              | Hidden tax amount                                         |
| string | base_hidden_tax_amount         | Base hidden tax amount                                    |
| string | shipping_hidden_tax_amount     | Shipping hidden tax amount                                |
| string | base_shipping_hidden_tax_amnt  | Base shipping hidden tax amount                           |
| string | shipping_incl_tax              | Shipping including tax                                    |
| string | base_shipping_incl_tax         | Base shipping including tax                               |
| string | base_customer_balance_amount   | Base customer balance amount                              |
| string | customer_balance_amount        | Customer balance amount                                   |
| string | bs_customer_bal_total_refunded | Refunded base customer balance amount                     |
| string | customer_bal_total_refunded    | Customer balance total refunded                           |
| string | base_gift_cards_amount         | Base gift cards amount                                    |
| string | gift_cards_amount              | Gift cards amount                                         |
| string | gw_base_price                  | Gift wrapping price refunded amount (using base currency) |
| string | gw_price                       | Gift wrapping price refunded amount                       |
| string | gw_items_base_price            | Gift wrapping items base price                            |
| string | gw_items_price                 | Gift wrapping items price                                 |
| string | gw_card_base_price             | Gift wrapping card base price                             |
| string | gw_card_price                  | Gift wrapping card price                                  |
| string | gw_base_tax_amount             | Gift wrapping tax amount refunded (using base currency)   |
| string | gw_tax_amount                  | Gift wrapping tax amount refunded                         |
| string | gw_items_base_tax_amount       | Gift wrapping items base tax amount                       |
| string | gw_items_tax_amount            | Gift wrapping items tax amount                            |
| string | gw_card_base_tax_amount        | Gift wrapping card base tax amount                        |
| string | gw_card_tax_amount             | Gift wrapping card tax amount                             |
| string | base_reward_currency_amount    | Base reward currency amount                               |
| string | reward_currency_amount         | Reward currency amount                                    |
|        |                                |                                                           |

|        |                              |                                            |
|--------|------------------------------|--------------------------------------------|
| string | reward_points_balance        | Reward points balance                      |
| string | reward_points_balance_refund | Reward points balance refund               |
| string | creditmemo_id                | Credit memo ID                             |
| array  | items                        | Array of salesOrderCreditmemoItemEntity    |
| array  | comments                     | Array of salesOrderCreditmemoCommentEntity |

The **salesOrderCreditmemoItemEntity** content is as follows:

| Type   | Name                              | Description                                             |
|--------|-----------------------------------|---------------------------------------------------------|
| string | item_id                           | Credit memo item ID                                     |
| string | parent_id                         | Parent ID                                               |
| string | weeee_tax_applied_row_amount      | Applied fixed product tax row amount                    |
| string | base_price                        | Base price                                              |
| string | base_weeee_tax_row_disposition    | Fixed product tax row disposition (in base currency)    |
| string | tax_amount                        | Tax amount                                              |
| string | base_weeee_tax_applied_amount     | Applied fixed product tax amount (in base currency)     |
| string | weeee_tax_row_disposition         | Fixed product tax row disposition                       |
| string | base_row_total                    | Base row total                                          |
| string | discount_amount                   | Discount amount                                         |
| string | row_total                         | Row total                                               |
| string | weeee_tax_applied_amount          | Applied fixed product tax amount                        |
| string | base_discount_amount              | Base discount amount                                    |
| string | base_weeee_tax_disposition        | Fixed product tax disposition (in base currency)        |
| string | price_incl_tax                    | Price including tax                                     |
| string | base_tax_amount                   | Base tax amount                                         |
| string | weeee_tax_disposition             | Fixed product tax disposition                           |
| string | base_price_incl_tax               | Base price including tax                                |
| string | qty                               | Quantity                                                |
| string | base_cost                         | Base cost                                               |
| string | base_weeee_tax_applied_row_amount | Applied fixed product tax row amount (in base currency) |
| string | price                             | Price                                                   |
| string | base_row_total_incl_tax           | Base row total including tax                            |
| string | row_total_incl_tax                | Row total including tax                                 |
| string | product_id                        | Product ID                                              |
| string | order_item_id                     | Order item ID                                           |

|        |                        |                           |
|--------|------------------------|---------------------------|
| string | additional_data        | Additional data           |
| string | description            | Description               |
| string | weee_tax_applied       | Applied fixed product tax |
| string | sku                    | Item SKU                  |
| string | name                   | Name                      |
| string | hidden_tax_amount      | Hidden tax amount         |
| string | base_hidden_tax_amount | Base hidden tax amount    |

The **salesOrderCreditmemoCommentEntity** content is as follows:

| Type   | Name                 | Description                                            |
|--------|----------------------|--------------------------------------------------------|
| string | parent_id            | Parent ID                                              |
| string | created_at           | Date of creation                                       |
| string | comment              | Comment data                                           |
| string | is_customer_notified | Defines whether the customer is notified               |
| string | comment_id           | Comment ID                                             |
| string | is_visible_on_front  | Defines whether the comment is visible on the frontend |

#### Faults:

| Fault Code | Fault Description                               |
|------------|-------------------------------------------------|
| 101        | Invalid filter given. Details in error message. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$filter = array(
 'order_id' => array(
 'or' => array(
 array(
 'from' => '673',
 'to' => '674',
),
 array(
 'from' => '677',
 'to' => NULL,
)
)
),
 'increment_id' => array(
 'or' => array(
 array(
 'from' => '100000617',
 'to' => '100000619',
),
)
)
);
```

```

 array(
 'from' => '100000619',
 'to' => NULL,
)
)
)
);
$ccreditmemoList = $proxy->call($sessionId, 'order_creditmemo.list', array($filter));

```

## Request Example SOAP V2

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderCreditmemoList($sessionId, '200000001');
var_dump($result);

```

## Request Example SOAP V2 (Complex Filter)

```

$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$complexFilter = array(
 'complex_filter' => array(
 array(
 'key' => 'state',
 'value' => array('key' => 'in', 'value' => '2,3')
)
)
);
$result = $client->salesOrderCreditmemoList($session, $complexFilter);

var_dump ($result);

```

## Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->salesOrderCreditmemoList((object)array('sessionId' => $sessionId->result));
var_dump($result->result);

```

## Response Example SOAP V1

```

array(
 0 => array(
 'store_id' => '1',
 'adjustment_positive' => NULL,
 'base_shipping_tax_amount' => '0.0000',
 'store_to_order_rate' => '1.0000',
 'base_discount_amount' => '0.0000',
 'base_to_order_rate' => '1.0000',
 'grand_total' => '60.0000',
 'base_adjustment_negative' => NULL,
)
)

```

```
'base_subtotal_incl_tax' => '55.0000',
'shipping_amount' => '5.0000',
'subtotal_incl_tax' => '55.0000',
'adjustment_negative' => NULL,
'base_shipping_amount' => '5.0000',
'store_to_base_rate' => '1.0000',
'base_to_global_rate' => '1.0000',
'base_adjustment' => '0.0000',
'base_subtotal' => '55.0000',
'discount_amount' => '0.0000',
'subtotal' => '55.0000',
'adjustment' => '0.0000',
'base_grand_total' => '60.0000',
'base_adjustment_positive' => NULL,
'base_tax_amount' => '0.0000',
'shipping_tax_amount' => '0.0000',
'tax_amount' => '0.0000',
'order_id' => '674',
'email_sent' => NULL,
'creditmemo_status' => NULL,
'state' => '2',
'shipping_address_id' => '1348',
'billing_address_id' => '1347',
'invoice_id' => NULL,
'cybersource_token' => NULL,
'store_currency_code' => 'USD',
'order_currency_code' => 'USD',
'base_currency_code' => 'USD',
'global_currency_code' => 'USD',
'transaction_id' => NULL,
'increment_id' => '100000617',
'created_at' => '2011-05-26 10:49:45',
'updated_at' => '2011-05-26 10:49:45',
'hidden_tax_amount' => '0.0000',
'base_hidden_tax_amount' => '0.0000',
'shipping_hidden_tax_amount' => NULL,
'base_shipping_hidden_tax_amnt' => NULL,
'shipping_incl_tax' => '5.0000',
'base_shipping_incl_tax' => '5.0000',
'base_customer_balance_amount' => NULL,
'customer_balance_amount' => NULL,
'bs_customer_bal_total_refunded' => '0.0000',
'customer_bal_total_refunded' => '0.0000',
'base_gift_cards_amount' => NULL,
'gift_cards_amount' => NULL,
'gw_base_price' => NULL,
'gw_price' => NULL,
'gw_items_base_price' => NULL,
'gw_items_price' => NULL,
'gw_card_base_price' => NULL,
'gw_card_price' => NULL,
'gw_base_tax_amount' => NULL,
'gw_tax_amount' => NULL,
'gw_items_base_tax_amount' => NULL,
'gw_items_tax_amount' => NULL,
'gw_card_base_tax_amount' => NULL,
'gw_card_tax_amount' => NULL,
'base_reward_currency_amount' => NULL,
'reward_currency_amount' => NULL,
'reward_points_balance' => NULL,
'reward_points_balance_refund' => NULL,
'creditmemo_id' => '617',
),
```

```
1 => array(
 'store_id' => '1',
 'adjustment_positive' => NULL,

 'creditmemo_id' => '620',
)
);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Sales Order Invoice

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

sales\_order\_invoice.addComment  
sales\_order\_invoice.cancel  
sales\_order\_invoice.capture  
sales\_order\_invoice.create  
sales\_order\_invoice.info  
sales\_order\_invoice.list

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

The Mage\_Sales module allows you to manage sales orders, invoices, shipments, and credit memos.

### Invoice

Allows you to manage invoices.

**Resource Name:** sales\_order\_invoice

#### Aliases:

- order\_invoice

#### Methods:

- [sales\\_order\\_invoice.list](#) - Retrieve a list of invoices using filters
- [sales\\_order\\_invoice.info](#) - Retrieve information about the invoice
- [sales\\_order\\_invoice.create](#) - Create a new invoice for an order
- [sales\\_order\\_invoice.addComment](#) - Add a new comment to an invoice
- [sales\\_order\\_invoice.capture](#) - Capture an invoice
- [sales\\_order\\_invoice.cancel](#) - Cancel an invoice

## Faults

| Fault Code | Fault Message                                    |
|------------|--------------------------------------------------|
| 100        | Requested shipment does not exists.              |
| 101        | Invalid filters given. Details in error message. |
| 102        | Invalid data given. Details in error message.    |
| 103        | Requested order does not exists                  |
|            |                                                  |

## Examples

### Example 1. Basic working with invoices.

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$notInvoicedOrderId = '100000003';

// Create new invoice
$newInvoiceId = $proxy->call($sessionId, 'sales_order_invoice.create',
array($notInvoicedOrderId, array(), 'Invoice Created', true, true));

// View new invoice
$invoice = $proxy->call($sessionId, 'sales_order_invoice.info', $newInvoiceId);

var_dump($invoice);

// Add Comment
$proxy->call($sessionId, 'sales_order_invoice.addComment', array($newInvoiceId,
'Invoice comment, some text', true, false));

// View invoice with new comment
$invoice = $proxy->call($sessionId, 'sales_order_invoice.info', $newInvoiceId);

var_dump($invoice);

$proxy->call($sessionId, 'sales_order_invoice.capture', $newInvoiceId);

// View captured invoice
$invoice = $proxy->call($sessionId, 'sales_order_invoice.info', $newInvoiceId);
var_dump($invoice);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Add Comment

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

[sales\\_order\\_invoice.addComm...](#)

[sales\\_order\\_invoice.cancel](#)

[sales\\_order\\_invoice.capture](#)

[sales\\_order\\_invoice.create](#)

[sales\\_order\\_invoice.info](#)

[sales\\_order\\_invoice.list](#)

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_invoice

#### Aliases:

- order\_invoice

#### Method:

- sales\_order\_invoice.addComment (SOAP V1)
- salesOrderInvoiceAddComment (SOAP V2)

Allows you to add a new comment to the order invoice.

#### Aliases:

- order\_invoice.addComment

#### Arguments:

| Type   | Name               | Description                              |
|--------|--------------------|------------------------------------------|
| string | sessionId          | Session ID                               |
| string | invoiceIncrementId | Invoice increment ID                     |
| string | comment            | Invoice comment (optional)               |
| int    | email              | Send invoice on email flag (optional)    |
| int    | includeComment     | Include comment in email flag (optional) |

#### Returns:

| Type | Description |
|------|-------------|
|      |             |

boolean

True if the comment is added to the invoice

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apikey');

$result = $client->call($session, 'sales_order_invoice.addComment',
array('invoiceIncrementId' => '200000006', 'comment' => 'invoice comment'));
var_dump ($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderInvoiceAddComment($sessionId, '200000006');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderInvoiceAddComment((object)array('sessionId' => $sessionId-
>result, 'invoiceIncrementId' => '200000006', 'comment' => 'invoice comment', 'email' =>
null, 'includeComment' => null));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Invoice Cancel

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

`sales_order_invoice.addComm...`

`sales_order_invoice.cancel`

`sales_order_invoice.capture`

`sales_order_invoice.create`

`sales_order_invoice.info`

`sales_order_invoice.list`

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_invoice

#### Aliases:

- `order_invoice`

#### Method:

- `sales_order_invoice.cancel (SOAP V1)`
- `salesOrderInvoiceCancel (SOAP V2)`

Allows you to cancel the required invoice. Note that not all order invoices can be canceled. Only some payment methods support canceling the order invoice (e.g., Google Checkout, PayPal Pro, PayPal Express Checkout).

#### Aliases:

- `order_invoice.cancel`

#### Arguments:

| Type   | Name               | Description          |
|--------|--------------------|----------------------|
| string | sessionId          | Session ID           |
| string | invoiceIncrementId | Invoice increment ID |

#### Returns:

| Type    | Description                            |
|---------|----------------------------------------|
| boolean | True if the order invoice is canceled. |

## Examples

## Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$invoiceIncrementId = '100000013';

$result = $proxy->call(
 $session,
 'sales_order_invoice.cancel',
 $invoiceIncrementId
);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Invoice Capture

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

`sales_order_invoice.addComm...`

`sales_order_invoice.cancel`

`sales_order_invoice.capture`

`sales_order_invoice.create`

`sales_order_invoice.info`

`sales_order_invoice.list`

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_invoice

### Aliases:

- order\_invoice

### Method:

- `sales_order_invoice.capture` (SOAP V1)
- `salesOrderInvoiceCapture` (SOAP V2)

Allows you to capture the required invoice. Note that not all order invoices can be captured. Only some payment methods support capturing the order invoice (e.g., PayPal Pro).

### Aliases:

- order\_invoice.capture

### Arguments:

| Type   | Name               | Description          |
|--------|--------------------|----------------------|
| string | sessionId          | Session ID           |
| string | invoiceIncrementId | Invoice increment ID |

### Returns:

| Type        | Description                                |
|-------------|--------------------------------------------|
| boolean\int | True (1) if the order invoice is captured. |

### Notes:

You should check the invoice to see if it can be captured before attempting to capture the invoice. Otherwise,

the API call will generate an error.

Invoices have states as defined in the model Mage\_Sales\_Model\_Order\_Invoice:

- STATE\_OPEN = 1
- STATE\_PAID = 2
- STATE\_CANCELED = 3

Also note that there is a method call in the model that checks this for you - canCapture(). And it also verifies that the payment can be captured, so the invoice state might not be the only condition that is required to allow it to be captured.

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$orderIncrementId = '100000016';

//Create invoice for order
$invoiceIncrementId = $proxy->call(
 $session,
 'sales_order_invoice.create',
 array(
 'orderIncrementId' => $orderIncrementId,
 array('order_item_id' => '15', 'qty' => '1')
)
);

//Capture invoice amount
$result = $proxy->call(
 $session,
 'sales_order_invoice.capture',
 $invoiceIncrementId
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$orderIncrementId = '100000016';

//Create invoice for order
$qty = array(
 array('order_item_id' => '15', 'qty' => '1')
);
$invoiceIncrementId = $proxy->salesOrderInvoiceCreate(
 $sessionId,
 $orderIncrementId,
 $qty);

//Capture invoice amount
$result = $proxy->salesOrderInvoiceCapture(
 $sessionId,
 $invoiceIncrementId
);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderInvoiceCapture((object)array('sessionId' => $sessionId-
>result, 'invoiceIncrementId' => '100000016'));

var_dump($result->result);
```

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Invoice Create

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

`sales_order_invoice.addComm...`  
`sales_order_invoice.cancel`  
`sales_order_invoice.capture`  
`sales_order_invoice.create`  
`sales_order_invoice.info`  
`sales_order_invoice.list`

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_invoice

#### Aliases:

- `order_invoice`

#### Method:

- `sales_order_invoice.create (SOAP V1)`
- `salesOrderInvoiceCreate (SOAP V2)`

Allows you to create a new invoice for an order.

#### Aliases:

- `order_invoice.create`

#### Arguments:

| Type   | Name               | Description                                            |
|--------|--------------------|--------------------------------------------------------|
| string | sessionId          | Session ID                                             |
| string | invoiceIncrementId | Order increment ID                                     |
| array  | itemsQty           | Array of orderItemIdQty (quantity of items to invoice) |
| string | comment            | Invoice comment (optional)                             |
| string | email              | Send invoice on email (optional)                       |
| string | includeComment     | Include comments in email (optional)                   |

#### Returns:

| Type   | Description               |
|--------|---------------------------|
| string | ID of the created invoice |

The **orderItemIdQty** content is as follows:

| Type   | Name          | Description   |
|--------|---------------|---------------|
| int    | order_item_id | Order item ID |
| double | qty           | Quantity      |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call(
 $session,
 'sales_order_invoice.create',
 array('orderIncrementId' => '200000008', array('15' => '1', '16' => '1'))
 // orderItemIdQty Array is Keyed with Order Item ID, with Value of qty to invoice
);
var_dump ($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

//Create invoice for order

// orderItemIdQty Array is Keyed with Order Item ID, with Value of qty to invoice
$qty = array('15' => '1', '16' => '1');

$invoiceIncrementId = $proxy->salesOrderInvoiceCreate(
 $sessionId,
 '200000008',
 $qty
);
var_dump($invoiceIncrementId);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderInvoiceCreate((object)array('sessionId' => $sessionId-
>result, 'orderIncrementId' => '200000008', 'itemsQty' => array('15' => '1', '16' =>
```

```
'1'), 'comment' => null,
'email' => null,
'includeComment' => null
));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Invoice Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

sales\_order\_invoice.addComm...  
sales\_order\_invoice.cancel  
sales\_order\_invoice.capture  
sales\_order\_invoice.create  
[sales\\_order\\_invoice.info](#)  
sales\_order\_invoice.list

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_invoice

#### Aliases:

- order\_invoice

#### Method:

- sales\_order\_invoice.info (SOAP V1)
- salesOrderInvoiceInfo (SOAP V2)

Allows you to retrieve information about the required invoice.

#### Aliases:

- order\_invoice.info

#### Arguments:

| Type   | Name               | Description          |
|--------|--------------------|----------------------|
| string | sessionId          | Session ID           |
| string | invoiceIncrementId | Invoice increment ID |

#### Returns:

| Type  | Name   | Description                      |
|-------|--------|----------------------------------|
| array | result | Array of salesOrderInvoiceEntity |

The **salesOrderInvoiceEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |                      |                                       |
|--------|----------------------|---------------------------------------|
| string | increment_id         | Increment ID                          |
| string | parent_id            | Parent ID                             |
| string | store_id             | Store ID                              |
| string | created_at           | Date of creation                      |
| string | updated_at           | Date of updating                      |
| string | is_active            | Defines whether the invoice is active |
| string | global_currency_code | Global currency code                  |
| string | base_currency_code   | Base currency code                    |
| string | store_currency_code  | Store currency code                   |
| string | order_currency_code  | Order currency code                   |
| string | store_to_base_rate   | Store to base rate                    |
| string | store_to_order_rate  | Store to order rate                   |
| string | base_to_global_rate  | Base to global rate                   |
| string | base_to_order_rate   | Base to order rate                    |
| string | subtotal             | Subtotal                              |
| string | base_subtotal        | Base subtotal                         |
| string | base_grand_total     | Base grand total                      |
| string | discount_amount      | Discount amount                       |
| string | base_discount_amount | Base discount amount                  |
| string | shipping_amount      | Shipping amount                       |
| string | base_shipping_amount | Base shipping amount                  |
| string | tax_amount           | Tax amount                            |
| string | base_tax_amount      | Base tax amount                       |
| string | billing_address_id   | Billing address ID                    |
| string | billing_firstname    | First name in the billing address     |
| string | billing_lastname     | Last name in the billing address      |
| string | order_id             | Order ID                              |
| string | order_increment_id   | Order increment ID                    |
| string | order_created_at     | Date of order creation                |
| string | state                | Order state                           |
| string | grand_total          | Grand total                           |
| string | invoice_id           | Invoice ID                            |
| array  | items                | Array of salesOrderInvoiceItemEntity  |

|       |          |                                         |
|-------|----------|-----------------------------------------|
| array | comments | Array of salesOrderInvoiceCommentEntity |
|-------|----------|-----------------------------------------|

The **salesOrderInvoiceItemEntity** content is as follows:

| Type   | Name                             | Description                                             |
|--------|----------------------------------|---------------------------------------------------------|
| string | increment_id                     | Increment ID                                            |
| string | parent_id                        | Parent ID                                               |
| string | created_at                       | Date of creation                                        |
| string | updated_at                       | Date of updating                                        |
| string | is_active                        | Active flag                                             |
| string | weee_tax_applied                 | Applied fixed product tax                               |
| string | qty                              | Quantity                                                |
| string | cost                             | Cost                                                    |
| string | price                            | Price                                                   |
| string | tax_amount                       | Tax amount                                              |
| string | row_total                        | Row total                                               |
| string | base_price                       | Base price                                              |
| string | base_tax_amount                  | Base tax amount                                         |
| string | base_row_total                   | Base row total                                          |
| string | base_weee_tax_applied_amount     | Applied fixed product tax amount (in base currency)     |
| string | base_weee_tax_applied_row_amount | Applied fixed product tax row amount (in base currency) |
| string | weee_tax_applied_amount          | Applied fixed product tax amount                        |
| string | weee_tax_applied_row_amount      | Applied fixed product tax row amount                    |
| string | weee_tax_disposition             | Fixed product tax disposition                           |
| string | weee_tax_row_disposition         | Fixed product tax row disposition                       |
| string | base_weee_tax_disposition        | Fixed product tax disposition (in base currency)        |
| string | base_weee_tax_row_disposition    | Fixed product tax row disposition (in base currency)    |
| string | sku                              | SKU                                                     |
| string | name                             | Name                                                    |
| string | order_item_id                    | Order item ID                                           |
| string | product_id                       | Product ID                                              |
| string | item_id                          | Item ID                                                 |

The **salesOrderInvoiceCommentEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |                      |                                          |
|--------|----------------------|------------------------------------------|
| string | increment_id         | Increment ID                             |
| string | parent_id            | Parent ID                                |
| string | created_at           | Date of creation                         |
| string | updated_at           | Date of updating                         |
| string | is_active            | Active flag                              |
| string | comment              | Invoice comment                          |
| string | is_customer_notified | Defines whether the customer is notified |
| string | comment_id           | Comment ID                               |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order_invoice.info', '200000006');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderInvoiceInfo($sessionId, '200000006');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderInvoiceInfo((object)array('sessionId' => $sessionId-
>result, 'invoiceIncrementId' => '200000006'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
 'store_id' => string '2' (length=1)
 'base_grand_total' => string '384.9800' (length=8)
 'shipping_tax_amount' => string '0.0000' (length=6)
 'tax_amount' => string '0.0000' (length=6)
```

```

'base_tax_amount' => string '0.0000' (length=6)
'store_to_order_rate' => string '1.0000' (length=6)
'base_shipping_tax_amount' => string '0.0000' (length=6)
'base_discount_amount' => string '0.0000' (length=6)
'base_to_order_rate' => string '1.0000' (length=6)
'grand_total' => string '384.9800' (length=8)
'shipping_amount' => string '5.0000' (length=6)
'subtotal_incl_tax' => string '379.9800' (length=8)
'base_subtotal_incl_tax' => string '379.9800' (length=8)
'store_to_base_rate' => string '1.0000' (length=6)
'base_shipping_amount' => string '5.0000' (length=6)
'total_qty' => string '1.0000' (length=6)
'base_to_global_rate' => string '1.0000' (length=6)
'subtotal' => string '379.9800' (length=8)
'base_subtotal' => string '379.9800' (length=8)
'discount_amount' => string '0.0000' (length=6)
'billing_address_id' => string '3' (length=1)
'is_used_for_refund' => null
'order_id' => string '2' (length=1)
'email_sent' => null
'can_void_flag' => string '0' (length=1)
'state' => string '2' (length=1)
'shipping_address_id' => string '4' (length=1)
'store_currency_code' => string 'USD' (length=3)
'transaction_id' => null
'order_currency_code' => string 'USD' (length=3)
'base_currency_code' => string 'USD' (length=3)
'global_currency_code' => string 'USD' (length=3)
'increment_id' => string '200000006' (length=9)
'created_at' => string '2012-03-30 12:02:19' (length=19)
'updated_at' => string '2012-03-30 12:02:19' (length=19)
'hidden_tax_amount' => string '0.0000' (length=6)
'base_hidden_tax_amount' => string '0.0000' (length=6)
'shipping_hidden_tax_amount' => string '0.0000' (length=6)
'base_shipping_hidden_tax_amnt' => null
'shipping_incl_tax' => string '5.0000' (length=6)
'base_shipping_incl_tax' => string '5.0000' (length=6)
'base_total_refunded' => null
'cybersource_token' => null
'invoice_id' => string '1' (length=1)
'order_increment_id' => string '200000002' (length=9)
'items' =>
 array
 0 =>
 array
 'parent_id' => string '1' (length=1)
 'base_price' => string '379.9800' (length=8)
 'tax_amount' => string '0.0000' (length=6)
 'base_row_total' => string '379.9800' (length=8)
 'discount_amount' => null
 'row_total' => string '379.9800' (length=8)
 'base_discount_amount' => null
 'price_incl_tax' => string '379.9800' (length=8)
 'base_tax_amount' => string '0.0000' (length=6)
 'base_price_incl_tax' => string '379.9800' (length=8)
 'qty' => string '1.0000' (length=6)
 'base_cost' => null
 'price' => string '379.9800' (length=8)
 'base_row_total_incl_tax' => string '379.9800' (length=8)
 'row_total_incl_tax' => string '379.9800' (length=8)
 'product_id' => string '1' (length=1)
 'order_item_id' => string '3' (length=1)
 'additional_data' => null

```

```
'description' => null
'sku' => string 'n2610-slider' (length=12)
'name' => string 'Nokia 2610 Phone' (length=16)
'hidden_tax_amount' => string '0.0000' (length=6)
'base_hidden_tax_amount' => string '0.0000' (length=6)
'base_weee_tax_applied_amount' => string '0.0000' (length=6)
'base_weee_tax_applied_row_amnt' => string '0.0000' (length=6)
'base_weee_tax_applied_row_amount' => string '0.0000' (length=6)
'weee_tax_applied_amount' => string '0.0000' (length=6)
'weee_tax_applied_row_amount' => string '0.0000' (length=6)
'weee_tax_applied' => string 'a:0:{}' (length=6)
'weee_tax_disposition' => string '0.0000' (length=6)
'weee_tax_row_disposition' => string '0.0000' (length=6)
'base_weee_tax_disposition' => string '0.0000' (length=6)
'base_weee_tax_row_disposition' => string '0.0000' (length=6)
'item_id' => string '1' (length=1)

'comments' =>
 array
 empty
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Invoice List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

`sales_order_invoice.addComm...`  
`sales_order_invoice.cancel`  
`sales_order_invoice.capture`  
`sales_order_invoice.create`  
`sales_order_invoice.info`  
`sales_order_invoice.list`

#### Sales Order Shipment ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_invoice

#### Aliases:

- order\_invoice

#### Method:

- sales\_order\_invoice.list (SOAP V1)
- salesOrderInvoiceList (SOAP V2)

Allows you to retrieve the list of order invoices. Additional filters can also be applied.

#### Aliases:

- order\_invoice.list

#### Arguments:

| Type   | Name      | Description                                          |
|--------|-----------|------------------------------------------------------|
| string | sessionId | Session ID                                           |
| array  | filters   | Array of filters for the list of invoices (optional) |

#### Returns:

| Type  | Name   | Description                      |
|-------|--------|----------------------------------|
| array | result | Array of salesOrderInvoiceEntity |

The **salesOrderInvoiceEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |                     |                                 |
|--------|---------------------|---------------------------------|
| string | increment_id        | Increment ID                    |
| string | created_at          | Date of invoice creation        |
| string | order_currency_code | Order currency code (e.g., EUR) |
| string | order_id            | Order ID                        |
| string | state               | Order state                     |
| string | grand_total         | Grand total amount invoiced     |
| string | invoice_id          | Invoice ID                      |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order_invoice.list');
var_dump($result);

// If you don't need the session anymore
//$client->endSession($session);
```

### Request Example SOAP V2 (List of All Invoices)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderInvoiceList($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (Complex Filter)

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$complexFilter = array(
 'complex_filter' => array(
 array(
 'key' => 'state',
 'value' => array('key' => 'in', 'value' => '2,3')
)
)
);
$result = $client->salesOrderInvoiceList($session, $complexFilter);

var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderInvoiceList((object)array('sessionId' => $sessionId-
>result));
var_dump($result->result);

```

## Response Example SOAP V1

```

array
0 =>
array
 'order_id' => string '2' (length=1)
 'increment_id' => string '200000001' (length=9)
 'created_at' => string '2012-03-30 12:02:19' (length=19)
 'state' => string '2' (length=1)
 'grand_total' => string '384.9800' (length=8)
 'order_currency_code' => string 'USD' (length=3)
 'invoice_id' => null
1 =>
array
 'order_id' => string '3' (length=1)
 'increment_id' => string '200000002' (length=9)
 'created_at' => string '2012-03-30 12:06:20' (length=19)
 'state' => string '2' (length=1)
 'grand_total' => string '339.9900' (length=8)
 'order_currency_code' => string 'USD' (length=3)
 'invoice_id' => null

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Sales Order Shipment

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

`sales_order_shipment.addCo...`  
`sales_order_shipment.addTrack`  
`sales_order_shipment.create`  
`sales_order_shipment.getCarri...`  
`sales_order_shipment.info`  
`sales_order_shipment.list`  
`sales_order_shipment.remove...`

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

The Mage\_Sales module allows you to manage sales orders, invoices, shipments, and credit memos.

### Shipment

Allows you to manage shipments and tracking numbers.

**Resource Name:** sales\_order\_shipment

#### Aliases:

- `order_shipment`

#### Methods:

- `sales_order_shipment.list` - Retrieve a list of shipments using filters
- `sales_order_shipment.info` - Retrieve information about the shipment
- `sales_order_shipment.create` - Create a new shipment for an order
- `sales_order_shipment.addComment` - Add a new comment to a shipment
- `sales_order_shipment.addTrack` - Add a new tracking number to a shipment
- `sales_order_shipment.removeTrack` - Remove tracking number from a shipment
- `sales_order_shipment.getCarriers` - Retrieve a list of allowed carriers for an order

### Faults

| Fault Code | Fault Message                                    |
|------------|--------------------------------------------------|
| 100        | Requested shipment not exists.                   |
| 101        | Invalid filters given. Details in error message. |
| 102        | Invalid data given. Details in error message.    |
| 103        | Requested order not exists.                      |

|     |                                                 |
|-----|-------------------------------------------------|
| 104 | Requested tracking not exists.                  |
| 105 | Tracking not deleted. Details in error message. |

## Examples

### Example 1. Basic working with shipments

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$notShippedOrderId = '100000003';

// Create new shipment
$newShipmentId = $proxy->call($sessionId, 'sales_order_shipment.create',
array($notShippedOrderId, array(), 'Shipment Created', true, true));

// View new shipment
$shipment = $proxy->call($sessionId, 'sales_order_shipment.info', $newShipmentId);

var_dump($shipment);

// Get allowed carriers for shipping
$allowedCarriers = $proxy->call($sessionId, 'sales_order_shipment.getCarriers',
$notShippedOrderId);

end($allowedCarriers);

$chooseCarrier = key($allowedCarriers);

var_dump($allowedCarriers);
var_dump($chooseCarrier);

// Add tracking
$newTrackId = $proxy->call($sessionId, 'sales_order_shipment.addTrack',
array($newShipmentId, $chooseCarrier, 'My Track', rand(5000, 9000)));

$shipment = $proxy->call($sessionId, 'sales_order_shipment.info', $newShipmentId);

var_dump($shipment);
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Shipment Add Comment

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

[sales\\_order\\_shipment.addCo...](#)

[sales\\_order\\_shipment.addTrack](#)

[sales\\_order\\_shipment.create](#)

[sales\\_order\\_shipment.getCarri...](#)

[sales\\_order\\_shipment.info](#)

[sales\\_order\\_shipment.list](#)

[sales\\_order\\_shipment.remove...](#)

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_shipment

#### Aliases:

- order\_shipment

#### Method:

- sales\_order\_shipment.addComment (SOAP V1)
- salesOrderShipmentAddComment (SOAP V2)

Allows you to add a new comment to the order shipment.

#### Aliases:

- order\_shipment.addComment

#### Arguments:

| Type   | Name                | Description                              |
|--------|---------------------|------------------------------------------|
| string | sessionId           | Session ID                               |
| string | shipmentIncrementId | Shipment increment ID                    |
| string | comment             | Shipment comment (optional)              |
| string | email               | Send email flag (optional)               |
| string | includeInEmail      | Include comment in email flag (optional) |

#### Returns:

| Type | Description |
|------|-------------|
|      |             |

boolean\int

True (1) if the comment is added to the order shipment

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order_shipment.addComment',
array('shipmentIncrementId' => '200000002', 'comment' => 'comment for the shipment',
'email' => null));
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderShipmentAddComment($sessionId, '200000002');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderShipmentAddComment((object)array('sessionId' =>
$sessionId->result, 'shipmentIncrementId' => '200000002', 'comment' => 'comment for
the shipment', 'email' => null, 'includeInEmail' => null));
var_dump($result->result);
```

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Shipment Add Track

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▲

[Sales Order](#) ▾

[Sales Order Credit Memo](#) ▾

[Sales Order Invoice](#) ▾

[Sales Order Shipment](#) ▾

[sales\\_order\\_shipment.addCo...](#)

[sales\\_order\\_shipment.addTrack](#)

[sales\\_order\\_shipment.create](#)

[sales\\_order\\_shipment.getCarri...](#)

[sales\\_order\\_shipment.info](#)

[sales\\_order\\_shipment.list](#)

[sales\\_order\\_shipment.remove...](#)

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

### Module: Mage\_Sales

Resource: sales\_order\_shipment

#### Aliases:

- order\_shipment

#### Method:

- sales\_order\_shipment.addTrack (SOAP V1)
- salesOrderShipmentAddTrack (SOAP V2)

Allows you to add a new tracking number to the order shipment.

#### Aliases:

- order\_shipment.addTrack

#### Arguments:

| Type   | Name                | Description                                     |
|--------|---------------------|-------------------------------------------------|
| string | sessionId           | Session ID                                      |
| string | shipmentIncrementId | Shipment increment ID                           |
| string | carrier             | Carrier code (ups, usps, dhl, fedex, or dhlint) |
| string | title               | Tracking title                                  |
| string | trackNumber         | Tracking number                                 |

#### Returns:

| Type | Description |
|------|-------------|
|      |             |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order_shipment.addTrack',
array('shipmentIncrementId' => '200000002', 'carrier' => 'ups', 'title' => 'tracking
title', 'trackNumber' => '123123'));
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if
necessary

$result = $proxy->salesOrderShipmentAddTrack($sessionId, '200000002', 'ups', 'tracking
title', '123123');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderShipmentAddTrack((object)array('sessionId' => $sessionId-
>result, 'shipmentIncrementId' => '200000002', 'carrier' => 'ups', 'title' =>
'tracking title', 'trackNumber' => '123123'));
var_dump($result->result);
```

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Shipment Create

[Edit this page on GitHub](#) 

## Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

## Sales ▾

### Sales Order ▾

### Sales Order Credit Memo ▾

### Sales Order Invoice ▾

### Sales Order Shipment ▾

`sales_order_shipment.addCo...`

`sales_order_shipment.addTrack`

`sales_order_shipment.create`

`sales_order_shipment.getCarri...`

`sales_order_shipment.info`

`sales_order_shipment.list`

`sales_order_shipment.remove...`

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_shipment

### Aliases:

- `order_shipment`

### Method:

- `sales_order_shipment.create (SOAP V1)`
- `salesOrderShipmentCreate (SOAP V2)`

Allows you to create a new shipment for an order.

### Aliases:

- `order_shipment.create`

### Arguments:

| Type   | Name             | Description                              |
|--------|------------------|------------------------------------------|
| string | sessionId        | Session ID                               |
| string | orderIncrementId | Order increment ID                       |
| array  | itemsQty         | Array of orderItemIdQty (optional)       |
| string | comment          | Shipment comment (optional)              |
| int    | email            | Send email flag (optional)               |
| int    | includeComment   | Include comment in email flag (optional) |

### Returns:

| Type   | Name                | Description           |
|--------|---------------------|-----------------------|
| string | shipmentIncrementId | Shipment increment ID |

The **orderItemIdQty** content is as follows:

| Type   | Name          | Description                     |
|--------|---------------|---------------------------------|
| int    | order_item_id | Order item ID                   |
| double | qty           | Quantity of items to be shipped |

**Notes:** The array of orderItemQty is used for partial shipment. To create shipment for all order items, you do not need to specify these attributes.

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');

$session = $proxy->login('apiUser', 'apiKey');

$orderIncrementId = '200000006';
$orderItemId = 3;
$qty = 5;
$itemQty = array(
 $orderItemId => $qty,
);

$result = $proxy->call(
 $session,
 'order_shipment.create',
 array(
 $orderIncrementId,
 $itemQty
)
);

var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$itemQty = array(
 array(
 'order_item_id' => 3,
 'qty' => 3
),
 array(
 'order_item_id' => 4,
 'qty' => 5
)
);

$result = $proxy->salesOrderShipmentCreate($sessionId, '200000006', $itemQty,
'shipment comment');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$itemQty = array(
 array(
 'order_item_id' => 3,
 'qty' => 3
),
 array(
 'order_item_id' => 4,
 'qty' => 5
)
);

$result = $proxy->salesOrderShipmentCreate((object)array(
 'sessionId' => $sessionId->result,
 'orderIncrementId' => '200000006',
 'itemsQty' => $itemQty,
 'comment' => 'shipment comment',
 'email' => null, 'includeComment' => null));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Shipment Get Carriers

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▲

[Sales Order](#) ▾

[Sales Order Credit Memo](#) ▾

[Sales Order Invoice](#) ▾

[Sales Order Shipment](#) ▲

[sales\\_order\\_shipment.addCo...](#)

[sales\\_order\\_shipment.addTrack](#)

[sales\\_order\\_shipment.create](#)

[sales\\_order\\_shipment.getCarri...](#)

[sales\\_order\\_shipment.info](#)

[sales\\_order\\_shipment.list](#)

[sales\\_order\\_shipment.remove...](#)

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Module: Mage\_Sales

Resource: sales\_order\_shipment

### Aliases:

- order\_shipment

### Method:

- sales\_order\_shipment.getCarriers (SOAP V1)
- salesOrderShipmentGetCarriers (SOAP V2)

Allows you to retrieve the list of allowed carriers for an order.

### Aliases:

- order\_shipment.getCarriers

### Arguments:

| Type   | Name             | Description        |
|--------|------------------|--------------------|
| string | sessionId        | Session ID         |
| string | orderIncrementId | Order increment ID |

### Returns:

| Type             | Name   | Description       |
|------------------|--------|-------------------|
| associativeArray | result | Array of carriers |

## Examples

## Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order_shipment.getCarriers', '2000000010');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderShipmentGetCarriers($sessionId, '2000000010');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderShipmentGetCarriers((object)array('sessionId' =>
$sessionId->result, 'orderIncrementId' => '2000000010'));
var_dump($result->result);
```

## Response Example SOAP V1

```
array
 'custom' => string 'Custom Value' (length=12)
 'dhl' => string 'DHL (Deprecated)' (length=16)
 'fedex' => string 'Federal Express' (length=15)
 'ups' => string 'United Parcel Service' (length=21)
 'usps' => string 'United States Postal Service' (length=28)
 'dhlint' => string 'DHL' (length=3)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Shipment Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

[sales\\_order\\_shipment.addCo...](#)

[sales\\_order\\_shipment.addTrack](#)

[sales\\_order\\_shipment.create](#)

[sales\\_order\\_shipment.getCarri...](#)

[sales\\_order\\_shipment.info](#)

[sales\\_order\\_shipment.list](#)

[sales\\_order\\_shipment.remove...](#)

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_shipment

#### Aliases:

- order\_shipment

#### Method:

- sales\_order\_shipment.info (SOAP V1)
- salesOrderShipmentInfo (SOAP V2)

Allows you to retrieve the shipment information.

#### Aliases:

- order\_shipment.info

#### Arguments:

| Type   | Name                | Description                 |
|--------|---------------------|-----------------------------|
| string | sessionId           | Session ID                  |
| string | shipmentIncrementId | Order shipment increment ID |

#### Returns:

| Type  | Name   | Description                       |
|-------|--------|-----------------------------------|
| array | result | Array of salesOrderShipmentEntity |

The **salesOrderShipmentEntity** content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |                     |                                          |
|--------|---------------------|------------------------------------------|
| string | increment_id        | Shipment increment ID                    |
| string | store_id            | Store ID                                 |
| string | created_at          | Date of shipment creation                |
| string | updated_at          | Date of shipment updating                |
| string | shipping_address_id | Shipping address ID                      |
| string | order_id            | Order ID                                 |
| string | total_qty           | Total quantity of items to ship          |
| string | shipment_id         | Shipment ID                              |
| array  | items               | Array of salesOrderShipmentItemEntity    |
| array  | tracks              | Array of salesOrderShipmentTrackEntity   |
| array  | comments            | Array of salesOrderShipmentCommentEntity |

The **salesOrderShipmentItemEntity** content is as follows:

| Type   | Name          | Description        |
|--------|---------------|--------------------|
| string | parent_id     | Parent ID          |
| string | sku           | Shipment item SKU  |
| string | name          | Shipment item name |
| string | order_item_id | Order item ID      |
| string | product_id    | Product ID         |
| string | weight        | Weight             |
| string | price         | Price              |
| string | qty           | Quantity of items  |
| string | item_id       | Item ID            |

The **salesOrderShipmentTrackEntity** content is as follows:

| Type   | Name         | Description                      |
|--------|--------------|----------------------------------|
| string | parent_id    | Parent ID                        |
| string | created_at   | Date of tracking number creation |
| string | updated_at   | Date of tracking number updating |
| string | carrier_code | Carrier code                     |
| string | title        | Track title                      |
| string | number       | Tracking number                  |
| string | order_id     | Order ID                         |
| string | track_id     | Track ID                         |

The **salesOrderShipmentCommentEntity** content is as follows:

| Type   | Name                 | Description                              |
|--------|----------------------|------------------------------------------|
| string | parent_id            | Parent ID                                |
| string | created_at           | Date of comment creation                 |
| string | comment              | Shipment comment text                    |
| string | is_customer_notified | Defines whether the customer is notified |
| string | comment_id           | Comment ID                               |

## Examples

### Request example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');
$session = $client->login('apiUser', 'apiKey');
$result = $client->call($session, 'sales_order_shipment.info', '200000003');
var_dump($result);
```

### Request example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderShipmentInfo($sessionId, '200000003');
var_dump($result);
```

### Request example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderShipmentInfo((object)array('sessionId' => $sessionId-
>result, 'shipmentIncrementId' => '200000003'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Shipment List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

#### Sales Order ▾

#### Sales Order Credit Memo ▾

#### Sales Order Invoice ▾

#### Sales Order Shipment ▾

`sales_order_shipment.addCo...`

`sales_order_shipment.addTrack`

`sales_order_shipment.create`

`sales_order_shipment.getCarri...`

`sales_order_shipment.info`

`sales\_order\_shipment.list`

`sales_order_shipment.remove...`

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_Sales

Resource: sales\_order\_shipment

#### Aliases:

- `order_shipment`

#### Method:

- `sales_order_shipment.list (SOAP V1)`
- `salesOrderShipmentList (SOAP V2)`

Allows you to retrieve the list of order shipments. Additional filters can be applied.

#### Aliases:

- `order_shipment.list`

#### Arguments:

| Type   | Name      | Description                                |
|--------|-----------|--------------------------------------------|
| string | sessionId | Session ID                                 |
| array  | filters   | Array of filters for the list of shipments |

Returns:

| Type  | Name   | Description                       |
|-------|--------|-----------------------------------|
| array | result | Array of salesOrderShipmentEntity |

The `salesOrderShipmentEntity` content is as follows:

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

|        |              |                                 |
|--------|--------------|---------------------------------|
| string | increment_id | Increment ID                    |
| string | created_at   | Date of shipment creation       |
| string | total_qty    | Total quantity of items to ship |
| string | shipment_id  | Shipment ID                     |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order_shipment.list');
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2 (List of All Shipments)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderShipmentList($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (Complex Filter)

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$complexFilter = array(
 'complex_filter' => array(
 array(
 'key' => 'created_at',
 'value' => array('key' => 'in', 'value' => '2012-03-30 12:54:46')
)
)
);
$result = $client->salesOrderShipmentList($session, $complexFilter);

var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));
```

```
$result = $proxy->salesOrderShipmentList((object)array('sessionId' => $sessionId->result));
var_dump($result->result);
```

## Response Example SOAP V1

```
array
0 =>
array
 'increment_id' => string '200000001' (length=9)
 'created_at' => string '2012-03-30 12:35:29' (length=19)
 'total_qty' => string '2.0000' (length=6)
 'shipment_id' => null
1 =>
array
 'increment_id' => string '200000002' (length=9)
 'created_at' => string '2012-03-30 12:54:46' (length=19)
 'total_qty' => string '1.0000' (length=6)
 'shipment_id' => null
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Remove Track

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▲

**Sales Order** ▾

**Sales Order Credit Memo** ▾

**Sales Order Invoice** ▾

**Sales Order Shipment** ▾

`sales_order_shipment.addCo...`

`sales_order_shipment.addTrack`

`sales_order_shipment.create`

`sales_order_shipment.getCarri...`

`sales_order_shipment.info`

`sales_order_shipment.list`

`sales_order_shipment.remove...`

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

## Module: Mage\_Sales

Resource: sales\_order\_shipment

### Aliases:

- `order_shipment`

### Method:

- `sales_order_shipment.removeTrack` (SOAP V1)
- `salesOrderShipmentRemoveTrack` (SOAP V2)

Allows you to remove a tracking number from the order shipment.

### Aliases:

- `order_shipment.removeTrack`

### Arguments:

| Type   | Name                | Description           |
|--------|---------------------|-----------------------|
| string | sessionId           | Session ID            |
| string | shipmentIncrementId | Shipment increment ID |
| string | trackId             | Track ID              |

### Returns:

| Type        | Description                                                  |
|-------------|--------------------------------------------------------------|
| boolean\int | True (1) if the tracking number is removed from the shipment |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'sales_order_shipment.removeTrack',
array('shipmentIncrementId' => '200000002', 'trackId' => '2'));
var_dump($result);

// If you don't need the session anymore
// $client->endSession($session);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->salesOrderShipmentRemoveTrack($sessionId, '200000002', '2');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->salesOrderShipmentRemoveTrack((object)array('sessionId' =>
$sessionId->result, 'shipmentIncrementId' => '200000002', 'trackId' => '2'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Enterprise Customer Balance

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Customer Balance](#) ▾

[Shopping Cart Customer Balance](#)

▼

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Store Credit API

Allows you to operate with customer virtual balance.

**Module:** Enterprise\_CustomerBalance

**Resource:** enterprise\_customerbalance

**Aliases:** storecredit

### Methods:

- [storecredit.balance](#) - Retrieve the customer store credit balance information
- [storecredit.history](#) - Retrieve the customer store credit history information

### Faults:

| Fault Code | Fault Message                               |
|------------|---------------------------------------------|
| 100        | Provided data is invalid.                   |
| 101        | No balance found with requested parameters. |
| 102        | No history found with requested parameters. |

**Resource:** enterprise\_customerbalance\_quote

**Aliases:** storecredit\_quote

### Methods:

- [storecredit\\_quote.setAmount](#) - Set amount from the customer store credit into a shopping cart (quote)
- [storecredit\\_quote.removeAmount](#) - Remove amount from a shopping cart (quote) and increase the

**Faults:**

| Fault Code | Fault Message                                            |
|------------|----------------------------------------------------------|
| 100        | Provided data is invalid.                                |
| 103        | No quote found with requested id.                        |
| 104        | Store credit can not be used for quote created by guest. |
| 105        | No store found with requested id or code.                |

**Example:**

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$customerId = 10002;
$websiteId = 1;
// Get customer balance amount
$balanceAmount = $proxy->call($sessionId, 'storecredit.balance', array($customerId,
$websiteId));
echo $balanceAmount . "
";
// Get store credit history for specified customer
$balanceHistory = $proxy->call($sessionId, 'storecredit.history', array($customerId));
print_r($balanceHistory);

$quoteId = 2;
$store = 'default';
// Set customer balance use in shopping cart (quote)
$balanceAmountUsedInQuote = $proxy->call($sessionId, 'storecredit_quote.setAmount',
array($quoteId, $store));
echo $balanceAmountUsedInQuote . "
";
// Unset customer balance use in shopping cart (quote)
$balanceAmountUsedInQuote = $proxy->call($sessionId, 'storecredit_quote.removeAmount',
array($quoteId));
echo $balanceAmountUsedInQuote;
```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Customer Balance

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

### Customer Balance ▾

storecredit.balance  
storecredit.history

### Shopping Cart Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Module: Enterprise\_CustomerBalance

Allows you to operate with customer virtual balance.

Resource:enterprise\_customerbalance

**Aliases:** storecredit

### Methods:

- [storecredit.balance](#) - Retrieve the customer store credit balance information
- [storecredit.history](#) - Retrieve the customer store credit history information

### Faults:

| Fault Code | Fault Message                               |
|------------|---------------------------------------------|
| 100        | Provided data is invalid.                   |
| 101        | No balance found with requested parameters. |
| 102        | No history found with requested parameters. |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Credit Balance

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

### Customer Balance ▾

[storecredit.balance](#)  
[storecredit.history](#)

### Shopping Cart Customer Balance ▾

## Enterprise Gift Card ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

**Module:** Customer Balance API

**Resource:** enterprise\_customerbalance

**Aliases:** `storecredit`

### Method:

- `enterprise_customerbalance.balance` (SOAP V1)
- `enterpriseCustomerbalanceBalance` (SOAP V2)

Allows you to retrieve the customer store credit balance amount.

### Arguments:

| Type   | Name       | Description |
|--------|------------|-------------|
| string | sessionId  | Session ID  |
| string | customerId | Customer ID |
| string | websiteId  | Website ID  |

### Return:

| Type         | Description            |
|--------------|------------------------|
| string/float | Virtual balance amount |

### Faults:

| Fault Code | Fault Message             |
|------------|---------------------------|
| 100        | Provided data is invalid. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$customerId = 4;
$websiteId = 2;

$balanceAmount = $proxy->call($sessionId, 'storecredit.balance', array($customerId,
$websiteId));
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if
necessary

$result = $proxy->enterpriseCustomerbalanceBalance($sessionId, '4', '2');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->enterpriseCustomerbalanceBalance((object)array('sessionId' =>
$sessionId->result, 'customerId' => '4', 'websiteId' => '2'));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Credit History

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Customer Balance** ▾

storecredit.balance

storecredit.history

**Shopping Cart Customer Balance** ▾

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

**Module:** Customer Balance API

**Resource:** enterprise\_customerbalance

**Aliases:** **storecredit**

**Method:**

- enterprise\_customerbalance.history (SOAP V1)
- enterpriseCustomerbalanceHistory (SOAP V2)

Allows you to retrieve the customer store credit history information.

**Arguments:**

| Type   | Name       | Description           |
|--------|------------|-----------------------|
| string | sessionId  | Session ID            |
| string | customerId | Customer ID           |
| string | websiteId  | Website ID (optional) |

**Return:**

| Type  | Name   | Description                                         |
|-------|--------|-----------------------------------------------------|
| array | result | Array of enterpriseCustomerbalanceHistoryItemEntity |

The **enterpriseCustomerbalanceHistoryItemEntity** content is as follows:

| Type   | Name       | Description     |
|--------|------------|-----------------|
| string | history_id | History item ID |

|        |                      |                                          |
|--------|----------------------|------------------------------------------|
| string | balance_id           | Customer balance ID                      |
| string | updated_at           | Date when the balance was updated        |
| string | action               | Code of the reason for balance change    |
| string | balance_amount       | Balance amount after changing            |
| string | balance_delta        | Balance change amount                    |
| string | additional_info      | Comment text                             |
| string | is_customer_notified | Defines whether the customer is notified |
| string | customer_id          | Balance owner ID                         |
| string | website_id           | Website ID                               |
| string | base_currency_code   | Currency code for the customer balance   |

#### Faults:

| Fault Code | Fault Message                               |
|------------|---------------------------------------------|
| 100        | Provided data is invalid.                   |
| 102        | No history found with requested parameters. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$customerId = 4;

$balanceHistory = $proxy->call($sessionId, 'storecredit.history', array($customerId));
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->enterpriseCustomerbalanceHistory($sessionId, '4');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->enterpriseCustomerbalanceHistory((object)array('sessionId' => $sessionId->result, 'customerId' => '4', 'websiteId' => '2'));
var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
 array
 'history_id' => string '2' (length=1)
 'balance_id' => string '1' (length=1)
 'updated_at' => string '2012-04-06 13:50:33' (length=19)
 'action' => string '1' (length=1)
 'balance_amount' => string '300.0000' (length=8)
 'balance_delta' => string '-200.0000' (length=9)
 'additional_info' => string 'By admin: john@example.com. (update balance)'
(length=44)
 'is_customer_notified' => string '0' (length=1)
 'customer_id' => string '4' (length=1)
 'website_id' => string '2' (length=1)
 'base_currency_code' => null
1 =>
 array
 'history_id' => string '1' (length=1)
 'balance_id' => string '1' (length=1)
 'updated_at' => string '2012-04-06 13:44:58' (length=19)
 'action' => string '2' (length=1)
 'balance_amount' => string '500.0000' (length=8)
 'balance_delta' => string '500.0000' (length=8)
 'additional_info' => string 'By admin: john@example.com. (new balance)'
(length=41)
 'is_customer_notified' => string '0' (length=1)
 'customer_id' => string '4' (length=1)
 'website_id' => string '2' (length=1)
 'base_currency_code' => null
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Shopping Cart Customer Balance

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Customer Balance](#) ▾

[Shopping Cart Customer Balance](#)

^

storecredit\_quote.removeAmount  
storecredit\_quote.setAmount

[Enterprise Gift Card](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Module: Enterprise\_CustomerBalance

Allows you to operate with customer virtual balance.

Resource: enterprise\_customerbalance\_quote

**Aliases:** storecredit\_quote

### Methods:

- [storecredit\\_quote.setAmount](#) - Set amount from the customer store credit into shopping cart (quote)
- [storecredit\\_quote.removeAmount](#) - Remove amount from the shopping cart (quote) and increase customer store credit

### Faults:

| Fault Code | Fault Message                                            |
|------------|----------------------------------------------------------|
| 100        | Provided data is invalid.                                |
| 103        | No quote found with requested id.                        |
| 104        | Store credit can not be used for quote created by guest. |
| 105        | No store found with requested id or code.                |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Balance Remove Amount

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Customer Balance** ▾

**Shopping Cart Customer Balance**

^

[storecredit\\_quote.removeAmount](#)

[storecredit\\_quote.setAmount](#)

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

**Module:** Customer Balance Quote API

**enterprise\_customerbalance\_quote**

**Aliases:** `storecredit_quote`

**Method:**

- `enterprise_customerbalance_quote.removeAmount` (SOAP V1)
- `shoppingCartCustomerbalanceRemoveAmount` (SOAP V2)

Allows you to remove store credit amount from the shopping cart (quote) and increase the customer store credit.

**Arguments:**

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | quotelId  | Shopping cart ID (quote ID)      |
| string | store     | Store view code or ID (optional) |

**Return:**

| Type    | Description                   |
|---------|-------------------------------|
| boolean | Result of removing the amount |

**Faults:**

| Fault Code | Fault Message             |
|------------|---------------------------|
| 100        | Provided data is invalid. |

|     |                                                          |
|-----|----------------------------------------------------------|
| 103 | No quote found with requested id.                        |
| 104 | Store credit can not be used for quote created by guest. |
| 105 | No store found with requested id or code.                |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$quoteId = 15;

$balanceAmountUsedInQuote = $proxy->call($sessionId, 'storecredit_quote.removeAmount',
array($quoteId));
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartCustomerbalanceRemoveAmount($sessionId, '15');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartCustomerbalanceRemoveAmount((object)array('sessionId' =>
$sessionId->result, 'quoteId' => '15', 'store' => null));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Balance Set Amount

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Customer Balance** ▾

**Shopping Cart Customer Balance**

^

storecredit\_quote.removeAmount

storecredit\_quote.setAmount

**Enterprise Gift Card** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

**Module:** Customer Balance Quote API

**enterprise\_customerbalance\_quote**

**Aliases:** **storecredit\_quote**

**Method:**

- enterprise\_customerbalance\_quote.setAmount (SOAP V1)
- shoppingCartCustomerbalanceSetAmount (SOAP V2)

Allows you to set amount from the customer store credit to the shopping cart.

**Arguments:**

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | quotelId  | Shopping cart ID (quote ID)      |
| string | store     | Store view code or ID (optional) |

**Return:**

| Type  | Description                                               |
|-------|-----------------------------------------------------------|
| float | Customer balance amount used in the shopping cart (quote) |

**Faults:**

| Fault Code | Fault Message             |
|------------|---------------------------|
| 100        | Provided data is invalid. |

|     |                                                          |
|-----|----------------------------------------------------------|
| 103 | No quote found with requested id.                        |
| 104 | Store credit can not be used for quote created by guest. |
| 105 | No store found with requested id or code.                |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$quoteId = 2;
$store = 'default';

$balanceAmountUsedInQuote = $proxy->call($sessionId, 'storecredit_quote.setAmount',
array($quoteId, $store));
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartCustomerbalanceSetAmount($sessionId, '15');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->shoppingCartCustomerbalanceSetAmount((object)array('sessionId' =>
$sessionId->result, 'quoteId' => '15', 'store' => null));
var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Enterprise Gift Card

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

### Cart Gift Card ▾

### Gift Card Account ▾

### Gift Card Customer ▾

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Customer Gift Card

Allows you to retrieve information on a gift card and redeem gift card amount.

**Resource Name:** giftcard\_customer

### Methods:

- [giftcard\\_customer.info](#) - Receive information about the gift card for a selected customer
- [giftcard\\_customer.redeem](#) - Redeem amount present on the gift card to the store credit

## Shopping Cart Gift Card

Allows you to retrieve, add, and remove a gift card from/to a shopping cart.

**Resource Name:** cart\_giftcard

### Methods:

- [cart\\_giftcard.list](#) - Retrieve the list of gift cards used in the shopping cart (quote).
- [cart\\_giftcard.add](#) - Add a gift card to a shopping cart (quote).
- [cart\\_giftcard.remove](#) - Remove a gift card from the shopping cart (quote).

## Gift Card Account

Allows you to create, update, remove, and retrieve information on a gift card account.

**Resource Name:** giftcard\_account

### Methods:

- [giftcard\\_account.create](#) - Create a new gift card
- [giftcard\\_account.list](#) - Get list of available gift cards
- [giftcard\\_account.update](#) - Update a gift card
-

[\*\*giftcard\\_account.info\*\*](#) - Receive full information about selected gift card

- [\*\*giftcard\\_account.remove\*\*](#) - Remove unnecessary gift card

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Cart Gift Card

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

#### Cart Gift Card ▾

[cart\\_giftcard.add](#)  
[cart\\_giftcard.list](#)  
[cart\\_giftcard.remove](#)

#### Gift Card Account ▾

#### Gift Card Customer ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: Enterprise\_GiftCard

Resource: cart\_giftcard

#### Methods:

- [cart\\_giftcard.list](#) - Retrieve the list of giftcards used in the shopping cart (quote).
- [cart\\_giftcard.add](#) - Add a giftcard to a shopping cart (quote).
- [cart\\_giftcard.remove](#) - Remove a giftcard from the shopping cart (quote).

#### Faults:

| Fault Code | Fault Message                                        |
|------------|------------------------------------------------------|
| 1001       | Can not make operation because store is not exists   |
| 1002       | Can not make operation because quote is not exists   |
| 1003       | Provided data is invalid                             |
| 1004       | Gift card account with requested code does not exist |
| 1005       | Error happened while adding gift card to quote       |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Gift Card Add

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Cart Gift Card** ▾

[cart\\_giftcard.add](#)

[cart\\_giftcard.list](#)

[cart\\_giftcard.remove](#)

**Gift Card Account** ▾

**Gift Card Customer** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

**Module:** GiftCard API

**Resource:** cart\_giftcard

**Method:**

- `cart_giftcard.add` (SOAP V1)
- `shoppingCartGiftcardAdd` (SOAP V2)

Allows you to add a gift card to the shopping cart (quote).

**Arguments:**

| Type   | Name                | Description                 |
|--------|---------------------|-----------------------------|
| string | sessionId           | Session ID                  |
| string | giftcardAccountCode | Gift card account code      |
| int    | quotId              | Shopping cart ID (quote ID) |
| int    | storeId             | Store ID (optional)         |

**Return:**

| Type    | Description                                 |
|---------|---------------------------------------------|
| boolean | True if the gift card is added to the quote |

**Faults:**

| Fault Code | Fault Message                                      |
|------------|----------------------------------------------------|
| 1001       | Can not make operation because store is not exists |
| 1002       | Can not make operation because quote is not exists |

|      |                                                      |
|------|------------------------------------------------------|
| 1004 | Gift card account with requested code does not exist |
| 1005 | Error happened while adding gift card to quote       |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$code = "giftcardAccountCode";
$quoteId = 15;

$giftcardList = $proxy->call(
 $sessionId,
 "cart_giftcard.add",
 array(
 $code,
 $quoteId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartGiftcardAdd($sessionId, 'giftcardAccountCode', '15');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->shoppingCartGiftcardAdd((object)array('sessionId' => $sessionId->result, 'giftcardAccountCode' => 'giftcardAccountCode', 'quoteId' => '15', 'storeId' => '3'));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Gift Card List

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

#### Cart Gift Card ▾

`cart_giftcard.add`  
`cart_giftcard.list`  
`cart_giftcard.remove`

#### Gift Card Account ▾

#### Gift Card Customer ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: GiftCard API

### Resource: cart\_giftcard

#### Method:

- `cart_giftcard.list` (SOAP V1)
- `shoppingCartGiftcardList` (SOAP V2)

Allows you to retrieve the list of gift cards used in the shopping cart (quote).

#### Arguments:

| Type   | Name      | Description                 |
|--------|-----------|-----------------------------|
| string | sessionId | Session ID                  |
| string | quotelId  | Shopping cart ID (quote ID) |
| string | storeId   | Store ID (optional)         |

#### Return:

| Type  | Name   | Description                                  |
|-------|--------|----------------------------------------------|
| array | result | Array of enterpriseGiftcardaccountListEntity |

The `enterpriseGiftcardaccountListEntity` content is as follows:

| Type   | Name               | Description                     |
|--------|--------------------|---------------------------------|
| string | giftcardaccount_id | Gift card account ID            |
| string | code               | Gift card code                  |
| string | used_amount        | Used amount (in store currency) |

|        |             |                                |
|--------|-------------|--------------------------------|
| string | base_amount | Base amount (in base currency) |
|--------|-------------|--------------------------------|

#### Faults:

| Fault Code | Fault Message                                      |
|------------|----------------------------------------------------|
| 1001       | Can not make operation because store is not exists |
| 1002       | Can not make operation because quote is not exists |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$quoteId = 15;

$giftcardList = $proxy->call(
 $sessionId,
 "cart_giftcard.list",
 array(
 $quoteId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartGiftcardList($sessionId, '15');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->shoppingCartGiftcardList((object)array('sessionId' => $sessionId->result, 'quoteId' => '15', 'storeId' => '3'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
 'giftcardaccount_id' => string '5' (length=1)
 'code' => string 'code' (length=12)
 'used_amount' => float 141.34
 'base_amount' => string '200.0000' (length=8)
1 =>
```

```
array
'giftcardaccount_id' => string '6' (length=1)
'code' => string 'code' (length=12)
'used_amount' => float 35.34
'base_amount' => string '50.0000' (length=7)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Gift Card Remove

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

#### Cart Gift Card ▾

`cart_giftcard.add`  
`cart_giftcard.list`  
`cart_giftcard.remove`

#### Gift Card Account ▾

#### Gift Card Customer ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: GiftCard API

### Resource: cart\_giftcard

#### Method:

- `cart_giftcard.remove` (SOAP V1)
- `shoppingCartGiftcardRemove` (SOAP V2)

Allows you to remove a gift card from the shopping cart (quote).

#### Arguments:

| Type   | Name                | Description                 |
|--------|---------------------|-----------------------------|
| string | sessionId           | Session ID                  |
| string | giftcardAccountCode | Gift card account code      |
| string | quotelId            | Shopping cart ID (quote ID) |
| string | storeId             | Store ID (optional)         |

#### Return:

| Type    | Description                                             |
|---------|---------------------------------------------------------|
| boolean | True if the gift card is removed from the shopping cart |

#### Faults:

| Fault Code | Fault Message                                      |
|------------|----------------------------------------------------|
| 1001       | Can not make operation because store is not exists |
| 1002       | Can not make operation because quote is not exists |

|      |                                                      |
|------|------------------------------------------------------|
| 1004 | Gift card account with requested code does not exist |
| 1005 | Error happened while adding gift card to quote       |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$code = "giftcardAccountCode";
$quoteId = 15;

$giftcardList = $proxy->call(
 $sessionId,
 "cart_giftcard.remove",
 array(
 $code,
 $quoteId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->shoppingCartGiftcardRemove($sessionId, 'giftcardAccountCode', '15');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->shoppingCartGiftcardRemove((object)array('sessionId' => $sessionId->result, 'giftcardAccountCode' => 'giftcardAccountCode', 'quoteId' => '15', 'storeId' => '3'));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Gift Card Account

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Cart Gift Card](#) ▾

[Gift Card Account](#) ▾

giftcard\_account.create

giftcard\_account.info

giftcard\_account.list

giftcard\_account.remove

giftcard\_account.update

[Gift Card Customer](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

## Module: Enterprise\_GiftCardAccount

Allows you to operate with giftcards.

Resource: giftcard\_account

### Methods:

- [giftcard\\_account.create](#) - Create a new giftcard
- [giftcard\\_account.list](#) - Get list of available giftcards
- [giftcard\\_account.update](#) - Update a giftcard
- [giftcard\\_account.info](#) - Receive full information about selected giftcard
- [giftcard\\_account.remove](#) - Remove unnecessary giftcard

### Faults:

| Fault Code | Fault Message                                        |
|------------|------------------------------------------------------|
| 100        | Gift card does not exists.                           |
| 101        | Invalid filters specified. Details in error message. |
| 102        | Unable to save data.                                 |
| 104        | Provided email notification data is invalid          |
| 105        | Provided gift card account data is invalid           |
| 106        | Gift card account with requested id does not exist   |
| 107        | Error occurs while deleting gift card                |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Account Create

[Edit this page on GitHub](#) 

**Introduction**

**Catalog** ▾

**Catalog Inventory** ▾

**Checkout** ▾

**Create Your Own API**

**Customer** ▾

**Directory** ▾

**Sales** ▾

**Enterprise Customer Balance** ▾

**Enterprise Gift Card** ▾

**Cart Gift Card** ▾

**Gift Card Account** ▾

[giftcard\\_account.create](#)  
[giftcard\\_account.info](#)  
[giftcard\\_account.list](#)  
[giftcard\\_account.remove](#)  
[giftcard\\_account.update](#)

**Gift Card Customer** ▾

**Enterprise Gift Message** ▾

**Miscellaneous** ▾

**WS-I Compliance**

**Module:** GiftCard API

**Resource:** giftcard\_account

**Method:**

- giftcard\_account.create (SOAP V1)
- giftcardAccountCreate (SOAP V2)

Allows you to create a new gift card account.

**Arguments:**

| Type   | Name                | Description                                               |
|--------|---------------------|-----------------------------------------------------------|
| string | sessionId           | Session ID                                                |
| array  | giftcardAccountData | Array of giftcardAccountCreateGiftcardAccountData         |
| array  | notificationData    | Array of giftcardAccountCreateNotificationData (optional) |

**Return:**

| Type   | Description                         |
|--------|-------------------------------------|
| string | ID of the created gift card account |

The **giftcardAccountCreateGiftcardAccountData** content is as follows:

| Type   | Name         | Description                                          |
|--------|--------------|------------------------------------------------------|
| string | status       | Gift card status: available, used, redeemed, expired |
| string | date_expires | Gift card expiration date in the YYYY-MM-DD format   |
| string | website_id   | Gift card website ID                                 |

|        |               |                                             |
|--------|---------------|---------------------------------------------|
| string | balance       | Initial gift card balance                   |
| string | state         | State: active or not active                 |
| string | is_redeemable | Defines whether the gift card is redeemable |

The **giftcardAccountCreateNotificationData** content is as follows:

| Type   | Name            | Description             |
|--------|-----------------|-------------------------|
| string | recipient_name  | Recipient name          |
| string | recipient_email | Recipient email address |
| string | recipient_store | Recipient store         |

#### Faults:

| Fault Code | Fault Message                               |
|------------|---------------------------------------------|
| 105        | Provided gift card account data is invalid. |
| 104        | Provided email notification data is invalid |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$giftcardToCreate = array(
 "status" => 'available',
 "is_redeemable" => 1,
 "balance" => 200,
 "website_id" => 2,
 "date_expires" => null
);

$giftcardId = $proxy->call(
 $sessionId,
 "giftcard_account.create",
 array(
 $giftcardToCreate
)
);

```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->giftcardAccountCreate($sessionId, array(
'status' => 'available',
'date_expires' => null,
'website_id' => '2',
'balance' => '200',
'state' => '1',
'is_redeemable' => '1'),

```

```

array(
'recipient_name' => 'name',
'recipient_email' => 'email',
'recipient_store' => null
);

var_dump($result);

```

### Request Example SOAP V2 (WS-I Compliance Mode)

```

$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->giftcardAccountCreate((object)array('sessionId' => $sessionId-
>result, 'giftcardAccountData' => array(
'status' => 'available',
'date_expires' => null,
'website_id' => '2',
'balance' => '200',
'state' => '1',
'is_redeemable' => '1'),
'notificationData' => array(
'recipient_name' => 'name',
'recipient_email' => 'email',
'recipient_store' => null
));

var_dump($result->result);

```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Account Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

#### Cart Gift Card ▾

#### Gift Card Account ▾

giftcard\_account.create  
[giftcard\\_account.info](#)  
giftcard\_account.list  
giftcard\_account.remove  
giftcard\_account.update

#### Gift Card Customer ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: GiftCard API

### Resource: giftcard\_account

#### Method:

- giftcard\_account.info (SOAP V1)
- giftcardAccountInfo (SOAP V2)

Allows you to receive full information about the selected gift card account.

#### Arguments:

| Type   | Name              | Description          |
|--------|-------------------|----------------------|
| string | sessionId         | Session ID           |
| string | giftcardAccountId | Gift card account ID |

#### Return:

| Type  | Name   | Description                    |
|-------|--------|--------------------------------|
| array | result | Array of giftcardAccountEntity |

The **giftcardAccountEntity** content is as follows:

| Type   | Name         | Description                            |
|--------|--------------|----------------------------------------|
| int    | giftcard_id  | Gift card account ID                   |
| string | code         | Gift card account code                 |
| int    | store_id     | Gift card account store ID             |
| string | date_created | Creation date in the YYYY-MM-DD format |

|        |               |                                                     |
|--------|---------------|-----------------------------------------------------|
| string | expire_date   | Expiration date in the YYYY-MM-DD format            |
| int    | is_active     | Defines whether the gift card account is active     |
| int    | is_redeemable | Defines whether the gift card account is redeemable |
| string | status        | Gift card account status                            |
| double | balance       | Gift card account initial balance                   |
| array  | history       | Array of giftcardAccountEntityHistory               |

The **giftcardAccountEntityHistory** content is as follows:

| Type   | Name          | Description    |
|--------|---------------|----------------|
| int    | record_id     | Record ID      |
| string | date          | Date           |
| string | action        | History action |
| double | balance_delta | Balance delta  |
| double | balance       | Balance        |
| string | info          | Record comment |

#### Faults:

| Fault Code | Fault Message              |
|------------|----------------------------|
| 100        | Gift card does not exists. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$giftcardId = 1;
$giftcardInfo = $proxy->call(
 $sessionId,
 "giftcard_account.info",
 array(
 $giftcardId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->giftcardAccountInfo($sessionId, '1');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->giftcardAccountInfo((object)array('sessionId' => $sessionId->result,
'giftcardAccountId' => '1'));

var_dump($result->result);
```

## Response Example SOAP V1

```
array
 'giftcard_id' => string '1' (length=1)
 'code' => string '02DQ9T1K0Y0V' (length=12)
 'store_id' => string '0' (length=1)
 'date_created' => string '2012-04-06' (length=10)
 'expire_date' => null
 'is_active' => string '0' (length=1)
 'status' => string 'Available' (length=9)
 'balance' => string '0.0000' (length=6)
 'is_redeemable' => string '1' (length=1)
 'history' =>
 array
 0 =>
 array
 'record_id' => string '1' (length=1)
 'date' => string '2012-04-06 14:31:57' (length=19)
 'action' => string 'Created' (length=7)
 'balance_delta' => string '0.0000' (length=6)
 'balance' => string '0.0000' (length=6)
 'info' => null
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Account List

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▲

[Cart Gift Card](#) ▾

[Gift Card Account](#) ▲

giftcard\_account.create  
giftcard\_account.info  
[giftcard\\_account.list](#)  
giftcard\_account.remove  
giftcard\_account.update

[Gift Card Customer](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

**Module:** GiftCard API

**Resource:** giftcard\_account

**Method:**

- giftcard\_account.list (SOAP V1)
- giftcardAccountList (SOAP V2)

Allows you to retrieve the list of available gift card accounts.

**Arguments:**

| Type   | Name      | Description                 |
|--------|-----------|-----------------------------|
| string | sessionId | Session ID                  |
| array  | filters   | Array of filters (optional) |

**Return:**

| Type  | Name   | Description                    |
|-------|--------|--------------------------------|
| array | result | Array of giftcardAccountEntity |

The **giftcardAccountEntity** content is as follows:

| Type   | Name         | Description                            |
|--------|--------------|----------------------------------------|
| int    | giftcard_id  | Gift card ID                           |
| string | code         | Gift card code                         |
| int    | store_id     | Gift card store ID                     |
| string | date_created | Creation date in the YYYY-MM-DD format |

|        |             |                                          |
|--------|-------------|------------------------------------------|
| string | expire_date | Expiration date in the YYYY-MM-DD format |
| int    | is_active   | Defines whether the gift card is active  |
| string | status      | Gift card status                         |
| double | balance     | Gift card balance                        |

#### Faults:

| Fault Code | Fault Message                                        |
|------------|------------------------------------------------------|
| 101        | Invalid filters specified. Details in error message. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$giftcardList = $proxy->call(
 $sessionId,
 "giftcard_account.list"
);
```

### Request Example SOAP V2 (List of All Gift Card Accounts)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->giftcardAccountList($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (Complex Filter)

```
$client = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

// If some stuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');
$complexFilter = array(
 'complex_filter' => array(
 array(
 'key' => 'state',
 'value' => array('key' => 'in', 'value' => '2,1')
)
)
);
$result = $client->giftcardAccountList($session, $complexFilter);

var_dump ($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');
```

```
$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->giftcardAccountList((object)array('sessionId' => $sessionId->result, 'filters' => null));

var_dump($result->result);
```

## Response Example SOAP V1

```
array
 0 =>
 array
 'giftcard_id' => string '1' (length=1)
 'code' => string 'code' (length=12)
 'store_id' => string '2' (length=1)
 'date_created' => string '2012-04-06' (length=10)
 'expire_date' => string '2012-04-22' (length=10)
 'is_active' => string '0' (length=1)
 'status' => string 'Available' (length=9)
 'balance' => string '300.0000' (length=8)
 1 =>
 array
 'giftcard_id' => string '2' (length=1)
 'code' => string 'code' (length=12)
 'store_id' => string '2' (length=1)
 'date_created' => string '2012-04-09' (length=10)
 'expire_date' => string '2012-04-25' (length=10)
 'is_active' => string '1' (length=1)
 'status' => string 'Available' (length=9)
 'balance' => string '500.0000' (length=8)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Account Remove

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▾

[Cart Gift Card](#) ▾

[Gift Card Account](#) ▾

giftcard\_account.create

giftcard\_account.info

giftcard\_account.list

[giftcard\\_account.remove](#)

giftcard\_account.update

[Gift Card Customer](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

Module: GiftCard API

Resource: giftcard\_account

### Method:

- giftcard\_account.remove (SOAP V1)
- giftcardAccountRemove (SOAP V2)

Allows you to remove the specified gift card account.

### Arguments:

| Type   | Name              | Description          |
|--------|-------------------|----------------------|
| string | sessionId         | Session ID           |
| string | giftcardAccountId | Gift card account ID |

### Return:

| Type    | Description                              |
|---------|------------------------------------------|
| boolean | True if the gift card account is removed |

### Faults:

| Fault Code | Fault Message                                      |
|------------|----------------------------------------------------|
| 106        | Gift card account with requested id does not exist |
| 107        | Error occurs while deleting gift card              |

## Examples

## Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$giftcardId = 2;
$result = $proxy->call(
 $sessionId,
 "giftcard_account.remove",
 array(
 $giftcardId
)
);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->giftcardAccountRemove($sessionId, '2');
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->giftcardAccountRemove((object)array('sessionId' => $sessionId->result, 'giftcardAccountId' => '2'));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Account Update

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▲

[Cart Gift Card](#) ▾

[Gift Card Account](#) ▲

[giftcard\\_account.create](#)  
[giftcard\\_account.info](#)  
[giftcard\\_account.list](#)  
[giftcard\\_account.remove](#)  
[giftcard\\_account.update](#)

[Gift Card Customer](#) ▾

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

**Module:** GiftCard API

**Resource:** giftcard\_account

**Method:**

- giftcard\_account.update (SOAP V1)
- giftcardAccountUpdate (SOAP V2)

Allows you to update the required gift card account.

**Arguments:**

| Type   | Name              | Description                            |
|--------|-------------------|----------------------------------------|
| string | sessionId         | Session ID                             |
| string | giftcardAccountId | Gift card account ID                   |
| array  | giftcardData      | Array of giftcardAccountEntityToUpdate |

**Return:**

| Type    | Description                              |
|---------|------------------------------------------|
| boolean | True is the gift card account is updated |

The **giftcardAccountEntityToUpdate** content is as follows:

| Type    | Name          | Description                                         |
|---------|---------------|-----------------------------------------------------|
| boolean | is_active     | Defines whether the gift card account is active     |
| boolean | is_redeemable | Defines whether the gift card account is redeemable |
| int     | store_id      | Store view ID                                       |

|        |              |                                          |
|--------|--------------|------------------------------------------|
| double | balance      | Gift card account initial balance        |
| string | date_expires | Expiration date in the YYYY-MM-DD format |

#### Faults:

| Fault Code | Fault Message        |
|------------|----------------------|
| 102        | Unable to save data. |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$giftcardId = 1;
$giftcardToUpdate = array(
 "balance" => 700
);

$result = $proxy->call(
 $sessionId,
 "giftcard_account.update",
 array(
 $giftcardId,
 $giftcardToUpdate
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->giftcardAccountUpdate($sessionId, '3', array(
'date_expires' => null,
'store_id' => '2',
'balance' => '700',
'is_active' => TRUE,
'is_redeemable' => TRUE));

var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->giftcardAccountUpdate((object)array('sessionId' => $sessionId,
>$result, 'giftcardAccountId' => '3', 'giftcardData' => array(
'balance' => '700',
'is_active' => TRUE,
'is_redeemable' => TRUE,
'store_id' => '2',
```

```
'date_expires' => null))));
var_dump($result->result);
```

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Gift Card Customer

[Edit this page on GitHub](#) 

[Introduction](#)

[Catalog](#) ▾

[Catalog Inventory](#) ▾

[Checkout](#) ▾

[Create Your Own API](#)

[Customer](#) ▾

[Directory](#) ▾

[Sales](#) ▾

[Enterprise Customer Balance](#) ▾

[Enterprise Gift Card](#) ▲

[Cart Gift Card](#) ▾

[Gift Card Account](#) ▾

[Gift Card Customer](#) ▲

giftcard\_customer.info

giftcard\_customer.redeem

[Enterprise Gift Message](#) ▾

[Miscellaneous](#) ▾

[WS-I Compliance](#)

### Module: Enterprise\_GiftCard

Resource: giftcard\_customer

#### Methods:

- [giftcard\\_customer.info](#) - Receive information about the giftcard for a selected customer
- [giftcard\\_customer.redeem](#) - Redeem amount present on the giftcard to the store credit

#### Faults:

| Fault Code | Fault Message                         |
|------------|---------------------------------------|
| 100        | Gift card does not exists.            |
| 101        | Gift card is not valid.               |
| 103        | Redemption functionality is disabled. |
| 104        | Unable to redeem gift card.           |

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Customer Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

#### Cart Gift Card ▾

#### Gift Card Account ▾

#### Gift Card Customer ▾

[giftcard\\_customer.info](#)

[giftcard\\_customer.redeem](#)

### Enterprise Gift Message ▾

### Miscellaneous ▾

### WS-I Compliance

## Module: GiftCard API

### Resource: giftcard\_customer

#### Method:

- giftcard\_customer.info (SOAP V1)
- giftcardCustomerInfo (SOAP V2)

Allows you to receive information about the gift card for a selected customer.

#### Arguments:

| Type   | Name      | Description    |
|--------|-----------|----------------|
| string | sessionId | Session ID     |
| string | code      | Gift card code |

#### Return:

| Type  | Name   | Description                     |
|-------|--------|---------------------------------|
| array | result | Array of giftcardCustomerEntity |

The **giftcardCustomerEntity** content is as follows:

| Type   | Name        | Description                                        |
|--------|-------------|----------------------------------------------------|
| double | balance     | Gift card balance                                  |
| string | expire_date | Gift card expiration date in the YYYY-MM-DD format |

#### Faults:

| Fault Code | Fault Message              |
|------------|----------------------------|
| 100        | Gift card does not exists. |
| 101        | Gift card is not valid.    |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$code = "code";

$giftcardInfo = $proxy->call(
 $sessionId,
 "giftcard_customer.info",
 array(
 $code
)
);

```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->giftcardCustomerInfo($sessionId, 'code');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->giftcardCustomerInfo((object)array('sessionId' => $sessionId->result, 'code' => 'code'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
 'balance' => string '500.0000' (length=8)
 'expire_date' => string '2012-04-25' (length=10)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Customer Redeem

[Edit this page on GitHub](#) 

## Introduction

## Catalog ▾

## Catalog Inventory ▾

## Checkout ▾

## Create Your Own API

## Customer ▾

## Directory ▾

## Sales ▾

## Enterprise Customer Balance ▾

## Enterprise Gift Card ▾

### Cart Gift Card ▾

### Gift Card Account ▾

### Gift Card Customer ▾

giftcard\_customer.info

[giftcard\\_customer.redeem](#)

## Enterprise Gift Message ▾

## Miscellaneous ▾

## WS-I Compliance

## Module: GiftCard API

### Resource: giftcard\_customer

#### Method:

- giftcard\_customer.redeem (SOAP V1)
- giftcardCustomerRedeem (SOAP V2)

Allows you to redeem amount from a giftcard to the customer store credit.

#### Arguments:

| Type   | Name        | Description    |
|--------|-------------|----------------|
| string | sessionId   | Session ID     |
| string | code        | Gift card code |
| string | customer_id | Customer ID    |
| string | store_id    | Store view ID  |

#### Return:

| Type    | Description                    |
|---------|--------------------------------|
| boolean | True if the amount is redeemed |

#### Faults:

| Fault Code | Fault Message              |
|------------|----------------------------|
| 100        | Gift card does not exists. |
| 101        | Gift card is not valid.    |

|     |                                       |
|-----|---------------------------------------|
| 103 | Redemption functionality is disabled. |
| 104 | Unable to redeem gift card.           |

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$code = "giftcardcode";
$customerId = 1;
$storeId = 1;

$giftcardInfo = $proxy->call(
 $sessionId,
 "giftcard_customer.redeem",
 array(
 $code,
 $customerId,
 $storeId
)
);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->giftcardCustomerRedeem($sessionId, 'giftcardcode', '1', '1');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->giftcardCustomerRedeem((object)array('sessionId' => $sessionId->result, 'code' => 'giftcardcode', 'customerId' => '1', 'storeId' => '1'));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Enterprise Gift Message

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

giftmessage.setForQuote

giftmessage.setForQuoteItem

giftmessage.setForQuoteProduct

### Miscellaneous ▾

### WS-I Compliance

## Module: Mage\_GiftMessage

This module allows you to set a gift message for a quote or for a product in a quote.

Resource: giftmessage

### Methods:

- [giftmessage.setForQuote](#) - Set a gift message for the shopping cart (quote)
- [giftmessage.setForQuoteItem](#) - Set a gift message for an item in the shopping cart (quote)
- [giftmessage.setForQuoteProduct](#) - Set a gift message for a product in the shopping cart (quote)

### Faults:

| Fault Code | Fault Message                                      |
|------------|----------------------------------------------------|
| 1001       | Can not make operation because store is not exists |
| 1002       | Can not make operation because quote is not exists |
| 1101       | Gift message data is not valid.                    |
| 1102       | Product's data is not valid.                       |
| 1103       | Quote item is not exists                           |

### Example:

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

// Create a quote, get quote identifier
$shoppingCartId = $proxy->call($sessionId, 'cart.create', array('magento_store'));

// set gift message for quote
```

```

$giftMessage = array (
 'from' => 'test',
 'to' => 'test',
 'message' => 'test'
);

$giftMessageSetForQuoteResult = $proxy->call($sessionId, "giftmessage.setForQuote",
array($shoppingCartId, $giftMessage, 'magento_store'));
print_r($giftMessageSetForQuoteResult);

// add products into shopping cart
$arrProducts = array(
 array(
 'product_id' => '1',
 'qty' => 2
),
 array(
 'sku' => 'testSKU',
 'qty' => 4
)
);
$resultCartProductAdd = $proxy->call($sessionId, 'cart_product.add',
array($shoppingCartId, $arrProducts));

// set gift message for product
$giftMessage = array (
 'from' => 'test for Product',
 'to' => 'test for Product',
 'message' => 'test for Product'
);

/**
 * Note:
 * array for product should be the same as for cart_product.add
 */
$giftMessageSetForProductResult = $proxy->call(
 $sessionId,
 "giftmessage.setForQuoteProduct",
 array(
 $shoppingCartId,
 array(
 array(
 'product' => $arrProducts[0],
 'giftmessage' => $giftMessage,
),
 array(
 'product' => $arrProducts[1],
 'giftmessage' => $giftMessage,
)
),
 'magento_store'
)
);

print_r($giftMessageSetForProductResult);

```

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Set For Quote

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

[giftmessage.setForQuote](#)

[giftmessage.setForQuoteItem](#)

[giftmessage.setForQuoteProduct](#)

### Miscellaneous ▾

### WS-I Compliance

## Module: GiftMessage API

### Resource: giftmessage

#### Method:

- `giftmessage.setForQuote` (SOAP V1)
- `giftMessageSetForQuote` (SOAP V2)

Allows you to set a global gift message for the shopping cart (quote).

#### Arguments:

| Type   | Name        | Description                      |
|--------|-------------|----------------------------------|
| string | sessionId   | Session ID                       |
| string | quotId      | Shopping cart ID (quote ID)      |
| array  | giftMessage | Array of giftMessageEntity       |
| string | store       | Store view ID or code (optional) |

#### Return:

| Type  | Name   | Description         |
|-------|--------|---------------------|
| array | result | giftMessageResponse |

The `giftMessageEntity` content is as follows:

| Type   | Name | Description            |
|--------|------|------------------------|
| string | from | Gift message sender    |
| string | to   | Gift message recipient |

|        |         |                   |
|--------|---------|-------------------|
| string | message | Gift message text |
|--------|---------|-------------------|

The **giftMessageResponse** content is as follows:

| Type    | Name     | Description                |
|---------|----------|----------------------------|
| string  | entityId | Entity ID                  |
| boolean | result   | Result of adding a message |
| string  | error    | Error                      |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'giftmessage.setForQuote', array('quoteId' => '10',
'giftMessage' => array('from' => 'John', 'to' => 'Sara', 'message' => 'Gift message
text')));
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->giftMessageSetForQuote($sessionId, 10, array(
'from' => 'John',
'to' => 'Sara',
'message' => 'Gift message text'
));

var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->giftMessageSetForQuote((object)array('sessionId' => $sessionId-
>result, 'quoteId' => '10', 'giftMessage' => array(
'from' => 'John',
'to' => 'Sara',
'message' => 'Gift message text'
)));

var_dump($result);
```

### Response Example SOAP V1

```
array
'entityId' => string '15' (length=2)
'result' => boolean true
'error' => string '' (length=0)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Set For Quote Item

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

giftmessage.setForQuote

[giftmessage.setForQuoteItem](#)

giftmessage.setForQuoteProduct

### Miscellaneous ▾

### WS-I Compliance

Module: GiftMessage API

Resource: giftmessage

#### Method:

- giftmessage.setForQuoteItem (SOAP V1)
- giftMessageSetForQuoteItem (SOAP V2)

Allows you to set a gift message for an item in the shopping cart (quote).

#### Arguments:

| Type   | Name        | Description                      |
|--------|-------------|----------------------------------|
| string | sessionId   | Session ID                       |
| string | quoteItemId | Shopping cart (quote) item ID    |
| array  | giftMessage | Array of giftMessageEntity       |
| string | store       | Store view ID or code (optional) |

#### Return:

| Type  | Name   | Description         |
|-------|--------|---------------------|
| array | result | giftMessageResponse |

The **giftMessageEntity** content is as follows:

| Type   | Name | Description            |
|--------|------|------------------------|
| string | from | Gift message sender    |
| string | to   | Gift message recipient |

|        |         |              |
|--------|---------|--------------|
| string | message | Gift message |
|--------|---------|--------------|

The **giftMessageResponse** content is as follows:

| Type    | Name     | Description                |
|---------|----------|----------------------------|
| string  | entityId | Entity ID                  |
| boolean | result   | Result of adding a message |
| string  | error    | Error                      |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'giftmessage.setForQuoteItem', array('quoteItemId' => '1', 'giftMessage' => array('from' => 'John', 'to' => 'Sara', 'message' => 'Gift message text')));
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->giftMessageSetForQuoteItem($sessionId, '1', array(
 'from' => 'John',
 'to' => 'Sara',
 'message' => 'Gift message text'
));

var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->giftMessageSetForQuoteItem((object)array('sessionId' => $sessionId->result, 'quoteItemId' => '1', 'giftMessage' => array(
 'from' => 'John',
 'to' => 'Sara',
 'message' => 'Gift message text'
)));

var_dump($result);
```

### Response Example SOAP V1

```
array
'entityId' => string '1' (length=1)
'result' => boolean true
'error' => string '' (length=0)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Set For Quote Product

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

giftmessage.setForQuote

giftmessage.setForQuoteItem

giftmessage.setForQuoteProduct

### Miscellaneous ▾

### WS-I Compliance

Module: GiftMessage API

Resource: giftmessage

#### Method:

- giftmessage.setForQuoteProduct (SOAP V1)
- giftMessageSetForQuoteProduct (SOAP V2)

Allows you to set a gift message for a product in the shopping cart (quote).

#### Arguments:

| Type   | Name                | Description                                   |
|--------|---------------------|-----------------------------------------------|
| string | sessionId           | Session ID                                    |
| string | quotelId            | Shopping cart ID (quote ID)                   |
| array  | productsAndMessages | Array of giftMessageAssociativeProductsEntity |
| string | store               | Store view ID or code (optional)              |

#### Return:

| Type  | Name   | Description                  |
|-------|--------|------------------------------|
| array | result | Array of giftMessageResponse |

The **giftMessageAssociativeProductsEntity** content is as follows:

| Type  | Name    | Description                        |
|-------|---------|------------------------------------|
| array | product | Array of shoppingCartProductEntity |
| array | message | Array of giftMessageEntity         |

The **giftMessageResponse** content is as follows:

| Type    | Name     | Description |
|---------|----------|-------------|
| string  | entityId | Entity ID   |
| boolean | result   | Result      |
| string  | error    | Error       |

The **shoppingCartProductEntity** content is as follows:

| Type             | Name              | Description                       |
|------------------|-------------------|-----------------------------------|
| string           | product_id        | ID of the product                 |
| string           | sku               | SKU of the product                |
| double           | qty               | Number of products                |
| associativeArray | options           | An array of options               |
| associativeArray | bundle_option     | An array of bundle item options   |
| associativeArray | bundle_option_qty | An array of bundle items quantity |
| ArrayOfString    | links             | An array of links                 |

The **giftMessageEntity** content is as follows:

| Type   | Name    | Description            |
|--------|---------|------------------------|
| string | from    | Gift message sender    |
| string | to      | Gift message recipient |
| string | message | Gift message           |

#### Faults:

No Faults.

## Examples

### Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->call($sessionId, 'giftmessage.setForQuoteProduct', array(10, array(
'product' => array(
'product_id' => '5',
'sku' => 'sku',
'qty' => '1',
'options' => null,
'bundle_option' => null,
'bundle_option_qty' => null,
'links' => null),
'message' => array(
'from' => 'Sara', 'to' => 'John', 'message' => 'Gift message text'
))));
```

```
var_dump($result);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login('apiUser', 'apiKey');

$result = $proxy->giftMessageSetForQuoteProduct($sessionId, '10', array(array(
'product' => array(
'product_id' => '5',
'sku' => 'sku',
'qty' => '1',
'options' => null,
'bundle_option' => null,
'bundle_option_qty' => null,
'links' => null),
'message' => array(
'from' => 'Sara', 'to' => 'John', 'message' => 'Gift message text'
)));
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' =>
'apiKey'));

$result = $proxy->giftMessageSetForQuoteProduct((object)array('sessionId' =>
$sessionId->result, 'quoteId' => '10', 'productsAndMessages' => array(array(
'product' => array(
'product_id' => '5',
'sku' => 'sku',
'qty' => '1',
'options' => null,
'bundle_option' => null,
'bundle_option_qty' => null,
'links' => null),
'message' => array(
'from' => 'Sara', 'to' => 'John', 'message' => 'Gift message text'
))));

var_dump($result->result);
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Miscellaneous

[Edit this page on GitHub](#) 

The following APIs allow you to access additional Magento information.

### Introduction

#### Catalog ▾

#### Catalog Inventory ▾

#### Checkout ▾

#### Create Your Own API

#### Customer ▾

#### Directory ▾

#### Sales ▾

#### Enterprise Customer Balance ▾

#### Enterprise Gift Card ▾

#### Enterprise Gift Message ▾

#### Miscellaneous ▾

[magento.info](#)

[store.info](#)

[store.list](#)

#### WS-I Compliance

## Module: Store View API

Allows you to retrieve information on the store view.

**Resource Name:** Store

### Methods:

- [store.info](#) - Get information about a store view
- [store.list](#) - Get the list of store views

## Module: Mage\_Core API

Allows you to get information about the current Magento installation.

**Resource Name:** core\_magento

**Aliases:** magento

### Method:

- [core\\_magento.info](#)

Create the Magento file system owner

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Magento Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

#### [magento.info](#)

store.info

store.list

### WS-I Compliance

## Magento Info API

Allows you to get information about the current Magento installation.

**Module:** Mage\_Core

**Resource:** core\_magento

**Aliases:** magento

### Method:

- core\_magento.info (SOAP V1)
- magentolInfo (SOAP V2)

Allows you to retrieve information about Magento version and edition.

**Aliases:** magento.info

### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

### Returns:

| Type   | Name            | Description                                           |
|--------|-----------------|-------------------------------------------------------|
| string | magento_version | Magento version                                       |
| string | magento_edition | Magento edition (Community, Professional, Enterprise) |

## Examples

## Request Example SOAP V1

```
$proxy = new SoapClient('http://magentohost/api/soap/?wsdl');
$sessionId = $proxy->login('apiUser', 'apiKey');
$magentoInfo = $proxy->call(
 $sessionId,
 'magento.info'
);
```

## Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->magentoInfo($sessionId);
var_dump($result);
```

## Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->magentoInfo((object)array('sessionId' => $sessionId->result));

var_dump($result->result);
```

## Response Example SOAP V1

```
array
 'magento_edition' => string 'Community' (length=9)
 'magento_version' => string '1.4.2.0-rc1' (length=11)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Store Info

[Edit this page on GitHub](#) 

### Introduction

### Catalog ▾

### Catalog Inventory ▾

### Checkout ▾

### Create Your Own API

### Customer ▾

### Directory ▾

### Sales ▾

### Enterprise Customer Balance ▾

### Enterprise Gift Card ▾

### Enterprise Gift Message ▾

### Miscellaneous ▾

[magento.info](#)

[store.info](#)

[store.list](#)

### WS-I Compliance

## Module: Store View API

Resource: store

### Method:

- `store.info` (SOAP V1)
- `storeInfo` (SOAP V2)

Allows you to retrieve information about the required store view.

### Arguments:

| Type   | Name      | Description                      |
|--------|-----------|----------------------------------|
| string | sessionId | Session ID                       |
| string | storeId   | Store view ID or code (optional) |

### Return:

| Type  | Name   | Description                       |
|-------|--------|-----------------------------------|
| array | result | Array of <code>storeEntity</code> |

The `storeEntity` content is as follows:

| Type   | Name       | Description     |
|--------|------------|-----------------|
| int    | store_id   | Store view ID   |
| string | code       | Store view code |
| int    | website_id | Website ID      |
| int    | group_id   | Group ID        |

|        |            |                                     |
|--------|------------|-------------------------------------|
| string | name       | Store name                          |
| int    | sort_order | Store view sort order               |
| int    | is_active  | Defines whether the store is active |

### Faults:

| Fault Code | Fault Message                   |
|------------|---------------------------------|
| 101        | Requested store view not found. |

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'store.info', '2');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->storeInfo($sessionId, '2');
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->storeInfo((object)array('sessionId' => $sessionId->result, 'storeId' => '2'));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
 'store_id' => string '2' (length=1)
 'code' => string 'english' (length=7)
 'website_id' => string '2' (length=1)
 'group_id' => string '2' (length=1)
 'name' => string 'English' (length=7)
 'sort_order' => string '0' (length=1)
 'is_active' => string '1' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## Store List

[Edit this page on GitHub](#) 

### Introduction

### Catalog

### Catalog Inventory

### Checkout

### Create Your Own API

### Customer

### Directory

### Sales

### Enterprise Customer Balance

### Enterprise Gift Card

### Enterprise Gift Message

### Miscellaneous

[magento.info](#)

[store.info](#)

[store.list](#)

### WS-I Compliance

## Module: Store View API

### Resource: store

#### Method:

- `store.list` (SOAP V1)
- `storeList` (SOAP V2)

Allows you to retrieve the list of store views.

#### Arguments:

| Type   | Name      | Description |
|--------|-----------|-------------|
| string | sessionId | Session ID  |

#### Return:

| Type  | Name   | Description          |
|-------|--------|----------------------|
| array | result | Array of storeEntity |

The `storeEntity` content is as follows:

| Type   | Name       | Description     |
|--------|------------|-----------------|
| int    | store_id   | Store view ID   |
| string | code       | Store view code |
| int    | website_id | Website ID      |
| int    | group_id   | Group ID        |
| string | name       | Store view name |

|     |            |                                     |
|-----|------------|-------------------------------------|
| int | sort_order | Store view sort order               |
| int | is_active  | Defines whether the store is active |

#### Faults:

No Faults

## Examples

### Request Example SOAP V1

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If somestuff requires API authentication,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'store.list');
var_dump ($result);
```

### Request Example SOAP V2

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl'); // TODO : change url
$sessionId = $proxy->login('apiUser', 'apiKey'); // TODO : change login and pwd if necessary

$result = $proxy->storeList($sessionId);
var_dump($result);
```

### Request Example SOAP V2 (WS-I Compliance Mode)

```
$proxy = new SoapClient('http://magentohost/api/v2_soap/?wsdl');

$sessionId = $proxy->login((object)array('username' => 'apiUser', 'apiKey' => 'apiKey'));

$result = $proxy->storeList((object)array('sessionId' => $sessionId->result));

var_dump($result->result);
```

### Response Example SOAP V1

```
array
0 =>
array
'store_id' => string '1' (length=1)
'code' => string 'default' (length=7)
'website_id' => string '1' (length=1)
'group_id' => string '1' (length=1)
'name' => string 'Default Store View' (length=18)
'sort_order' => string '0' (length=1)
'is_active' => string '1' (length=1)
1 =>
array
'store_id' => string '2' (length=1)
'code' => string 'english' (length=7)
'website_id' => string '2' (length=1)
'group_id' => string '2' (length=1)
'name' => string 'English' (length=7)
```

```
'sort_order' => string '0' (length=1)
'is_active' => string '1' (length=1)
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## WS-I Compliance

[Edit this page on GitHub](#) 

### Introduction

### Catalog

### Catalog Inventory

### Checkout

### Create Your Own API

### Customer

### Directory

### Sales

### Enterprise Customer Balance

### Enterprise Gift Card

### Enterprise Gift Message

### Miscellaneous

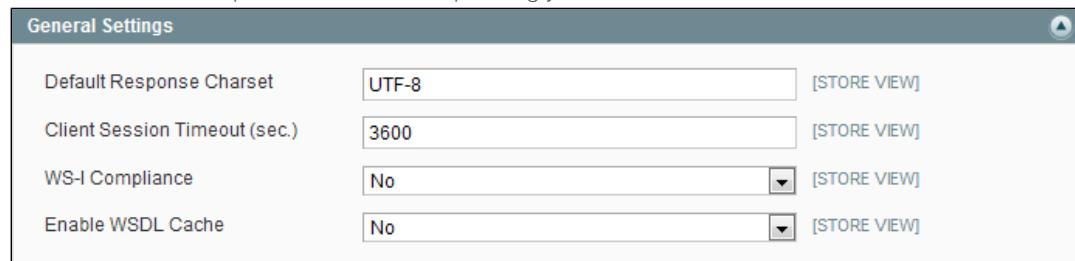
### WS-I Compliance

## WS-I Compliance Mode

Magento provides you with the ability to use two modes for SOAP API V2. These are with WS-I compliance mode enabled and WS-I compliance mode disabled. The first one was introduced to make the system flexible, namely, to increase compatibility with .NET and Java programming languages.

To enable/disable the WS-I compliance mode, perform the following steps:

1. In the Magento Admin Panel, go to **System > Configuration > Magento Core API**.
2. In the WS-I Compliance drop-down, select **Yes** to enable the WS-I compliance mode and **No** to disable the WS-I compliance mode, correspondingly.



| General Settings              |                                    |
|-------------------------------|------------------------------------|
| Default Response Charset      | UTF-8 <a href="#">[STORE VIEW]</a> |
| Client Session Timeout (sec.) | 3600 <a href="#">[STORE VIEW]</a>  |
| WS-I Compliance               | No <a href="#">[STORE VIEW]</a>    |
| Enable WSDL Cache             | No <a href="#">[STORE VIEW]</a>    |

The WS-I compliant mode uses the same WSDL endpoint as SOAP API V2 does. The key difference is that XML namespaces are used in WS-I compliance mode.

WSDL file with disabled WS-I compliance mode:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<definitions xmlns:typens="urn:Magento" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
 xmlns="http://schemas.xmlsoap.org/wsdl/" name="Magento" targetNamespace="urn:Magento">
 <types>
 <schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:Magento">
 <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
 schemaLocation="http://schemas.xmlsoap.org/soap/encoding/"/>
 <complexType name="associativeEntity">
 <all>
 <element name="key" type="xsd:string"/>
 <element name="value" type="xsd:string"/>
 </all>
 </complexType>
 <complexType name="associativeArray">
 <complexContent>
 <restriction base="soapenc:Array">
 <attribute ref="soapenc:arrayType" wsdl:arrayType="typens:associativeEntity[]"/>
 </restriction>
 </complexContent>
 </complexType>
 <complexType name="associativeMultiEntity">
```

WSDL file with enabled WS-I compliance mode:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<wsdl:definitions xmlns:typens="urn:Magento" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
 name="Magento" targetNamespace="urn:Magento">
 <wsdl:types>
 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:Magento">
 <xsd:complexType name="associativeEntity">
 <xsd:sequence>
 <xsd:element name="key" type="xsd:string"/>
 <xsd:element name="value" type="xsd:string"/>
 </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="associativeArray">
 <xsd:sequence>
 <xsd:element minOccurs="0" maxOccurs="unbounded" name="complexObjectArray"
 type="typens:associativeEntity"/>
 </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="associativeMultiEntity">
 <xsd:sequence>
 <xsd:element name="key" type="xsd:string"/>
 <xsd:element name="value" type="typens:ArrayOfString"/>
 </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="associativeMultiArray">
```

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020.

Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Upgrading to and Verifying Magento Community Edition and Enterprise Edition —Part 2

[Edit this page on GitHub](#)



## Table of Contents

- [Overview](#)
- [Upgrade Roadmaps](#)
- [Getting Ready For Your Upgrade](#)
- [Upgrading the File System, Solr, and Database](#)
- [Running the URL Redirect Script \*EE 1.13.0.0 and 1.13.0.1 only\*](#)
- [Completing the Magento Upgrade](#)
- [Testing the Magento Upgrade](#)
- [Setting Magento File System Permissions and Ownership After Upgrade](#)
- [Troubleshooting](#)

## Overview

This article discusses how you upgrade to:

- Magento Community Edition (CE) 1.8 or 1.9.x

- Magento Enterprise Edition (EE) 1.13.0.2, 1.13.1, or 1.14.x



**Important:** Magento recommends CE 1.9.1.0 or later or EE 1.14.1.0 or later for *all CE and EE installations and upgrades* to get the latest fixes, features, and security updates.

Because of changes to URL rewrites, the following upgrades are more complex than other upgrades.

- Upgrading from EE 1.12 or earlier to EE 1.14.x
- Upgrading from EE 1.13.0.0 or EE 1.13.0.1 to EE 1.14.x



**Note:** If you're upgrading from EE 1.13.0.2 to EE 1.13.1.0 or later, your upgrade doesn't involve changes to URL rewrites and you can skip many of the steps discussed in the other upgrades. Continue with [Getting Ready For Your Upgrade](#).

## Upgrade Roadmaps

- High-level roadmaps for upgrading to EE 1.13.0.2 or later are discussed in [Understanding the Upgrade to EE 1.13.0.2 or EE 1.14](#).
- To upgrade from EE 1.13.0.2 to EE 1.13.1.0 or later (including EE 1.14.x), see [Getting Ready For Your Upgrade](#).
- To upgrade to CE 1.8 or 1.9, see [Upgrade Roadmap for CE 1.8 or 1.9](#)

## Getting Ready For Your Upgrade

This section discusses how to get ready for your upgrade by backing up the database and customizations on the file system. The steps that follow do not affect your current production system. You can continue serving customers with no downtime.

To get ready for your upgrade:

1. Get the Magento installation archive as discussed in [Getting Magento CE or EE](#).
2. Install Magento in a different directory:
  - **Recommended.** Set up a new system (that is, another host) on which to install Magento. The system should be identical to, if not better than, your current system. The new system must meet the [Magento system requirements](#).
  - Install Magento in a new, empty root installation directory on the same server.



**Important:** *Do not* upgrade Magento in the same directory on the same server because post-upgrade errors are likely to occur.

- Magento **strongly** recommends you observe the following guidelines when you set up the Magento database in your development environment:
  - Magento for the first time uses [MySQL database triggers](#) to improve database access during reindexing. Magento does not support any custom triggers in the Magento database because custom triggers can introduce incompatibilities with future Magento versions.
  - Familiarize yourself with [these potential MySQL trigger limitations](#) before you continue.
  - If you use MySQL database replication, Magento **does not** support MySQL statement-

based replication. Make sure you use only [row-based replication](#).

- Review the MySQL recommendations for [MySQL password security](#).
- Complete the tasks discussed in the following sections in the order shown:
  - [Pre-Upgrade Tasks For Your Production Environment](#)
  - [Pre-Upgrade Tasks For Your Development Environment](#)

## Pre-Upgrade Tasks For Your Production Environment

Complete the tasks discussed in the following sections in the order shown:

- [Enabling Exception Logging](#)
- [Disabling cron jobs \(\*EE 1.13.0.2 or later\*\)](#)
- [Setting All Indexers for Update on Save](#)
- [Flushing the Cache](#)
- [Backing Up the Database](#)
- [Archiving Customizations and Extensions](#)
- [Restoring Configuration Settings](#)

### Enabling Exception Logging

Before you start your upgrade, you should enable exception logging so it will be enabled on the development system. Without exception logging, it will be more difficult to diagnose errors during your upgrade. You can disable exception logging after you've exported the Magento database.

To enable exception logging:

1. Log in to the Admin Panel as an administrator.
2. Click **System > Configuration**.
3. In the ADVANCED group, click **Developer**.  
This enables you to enable exception logging so you'll see exceptions in the development environment after the upgrade.
4. In the right pane, click **Log Settings** to expand it.

The following figure shows an example.

|                                                                            |                                            |                    |
|----------------------------------------------------------------------------|--------------------------------------------|--------------------|
| <b>Developer</b>                                                           |                                            | <b>Save Config</b> |
| <b>Developer Client Restrictions</b>                                       |                                            |                    |
| Debug                                                                      |                                            |                    |
| Template Settings                                                          |                                            |                    |
| Translate Inline                                                           |                                            |                    |
| <b>Log Settings</b>                                                        |                                            |                    |
| Enabled                                                                    | <input type="text" value="Yes"/>           | [STORE VIEW]       |
| System Log File Name                                                       | <input type="text" value="system.log"/>    | [STORE VIEW]       |
| Logging from Mage::log(). File is located in {{base_dir}}/var/log          |                                            |                    |
| Exceptions Log File Name                                                   | <input type="text" value="exception.log"/> | [STORE VIEW]       |
| Logging from Mage::logException(). File is located in {{base_dir}}/var/log |                                            |                    |
| <b>JavaScript Settings</b>                                                 |                                            |                    |
| <b>CSS Settings</b>                                                        |                                            |                    |

5. From the **Enabled** list, click **Yes**.

6. Optionally change the names of the log files.

7. Click **Save Config**.

## Disabling cron Jobs

Before upgrading to EE 1.13.0.2 or later, disable all running cron jobs. A simple way to stop cron jobs follows; consult an appropriate reference for other options.

As a user with `root` privileges, enter one of the following commands:

- Ubuntu: `service crond stop`
- CentOS: `service cron stop`

## Setting All Indexers for Update on Save

*This section applies to upgrading Magento EE only.* Skip this section if you're using Magento CE.

Because of the changes to indexing in Magento EE 1.13, you must set all indexers to update on save before you upgrade; otherwise, unpredictable performance will result. You can revert indexer settings after you've exported the Magento database.

To set indexers to update on save:

1. Click **System > Index Management**.
2. On the top left corner of the row, click **Select All** to select all indexers.
3. From the **Options** list, click **Change indexing mode**.  
The **Index Mode** list displays.
4. From the **Index Mode** list, click **Update on Save**.

The following figure shows an example.

| Index Management                                             |                                                          |                |        |                 |                         |              |
|--------------------------------------------------------------|----------------------------------------------------------|----------------|--------|-----------------|-------------------------|--------------|
| Index                                                        | Description                                              | Mode           | Status | Update Required | Updated At              | Action       |
| [ <input checked="" type="checkbox"/> ] Product Attributes   | Index product attributes for layered navigation building | Update on Save | READY  | NO              | Mar 2, 2013 4:18:26 AM  | Reindex Data |
| [ <input checked="" type="checkbox"/> ] Product Prices       | Index product prices                                     | Update on Save | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input checked="" type="checkbox"/> ] Catalog URL Rewrites | Index product and categories URL rewrites                | Manual Update  | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input checked="" type="checkbox"/> ] Product Flat Data    | Reorganize EAV product structure to flat structure       | Manual Update  | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input checked="" type="checkbox"/> ] Category Flat Data   | Reorganize EAV category structure to flat structure      | Manual Update  | READY  | NO              | Feb 24, 2013 7:01:21 AM | Reindex Data |
| [ <input checked="" type="checkbox"/> ] Category Products    | Indexed category/products association                    | Update on Save | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input checked="" type="checkbox"/> ] Catalog Search Index | Rebuild Catalog product fulltext search index            | Manual Update  | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input checked="" type="checkbox"/> ] Stock Status         | Index Product Stock Status                               | Update on Save | READY  | NO              | Mar 2, 2013 4:18:26 AM  | Reindex Data |
| [ <input checked="" type="checkbox"/> ] Tag Aggregation Data | Rebuild Tag aggregation data                             | Update on Save | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |

5. At the top right corner, click **Submit**.

6. Make sure all indexers are now set for Update on Save as the following figure shows.

| Index Management                                  |                                                          |                |        |                 |                         |              |
|---------------------------------------------------|----------------------------------------------------------|----------------|--------|-----------------|-------------------------|--------------|
| Index                                             | Description                                              | Mode           | Status | Update Required | Updated At              | Action       |
| [ <input type="checkbox"/> ] Product Attributes   | Index product attributes for layered navigation building | Update on Save | READY  | NO              | Mar 2, 2013 4:18:26 AM  | Reindex Data |
| [ <input type="checkbox"/> ] Product Prices       | Index product prices                                     | Update on Save | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input type="checkbox"/> ] Catalog URL Rewrites | Index product and categories URL rewrites                | Update on Save | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input type="checkbox"/> ] Product Flat Data    | Reorganize EAV product structure to flat structure       | Update on Save | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input type="checkbox"/> ] Category Flat Data   | Reorganize EAV category structure to flat structure      | Update on Save | READY  | NO              | Feb 24, 2013 7:01:21 AM | Reindex Data |
| [ <input type="checkbox"/> ] Category Products    | Indexed category/products association                    | Update on Save | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input type="checkbox"/> ] Catalog Search Index | Rebuild Catalog product fulltext search index            | Update on Save | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |
| [ <input type="checkbox"/> ] Stock Status         | Index Product Stock Status                               | Update on Save | READY  | NO              | Mar 2, 2013 4:18:26 AM  | Reindex Data |
| [ <input type="checkbox"/> ] Tag Aggregation Data | Rebuild Tag aggregation data                             | Update on Save | READY  | NO              | Mar 2, 2013 4:18:27 AM  | Reindex Data |

## Flushing the Cache

Clear the Magento cache as follows:

1. Log in to the Admin Panel as an administrator.
2. Click **System > Cache Management**.
3. Click **Flush Magento Cache**.

## Backing Up the Database

Back up your database using `mysqldump` or another tool. `mysqldump` syntax follows:

```
mysqldump -u root magento-database-name > export-file-name.sql
```

## Archiving Customizations and Extensions

To archive your custom themes and extensions:

1. Archive the Magento `media` directory and all subdirectories. For example,

```
cd [your Magento install dir]
tar -czf media.tgz media
```

2. Archive any custom themes in the following directories:

```
[your Magento install dir]/app/design/frontend
[your Magento install dir]/skin/frontend
```

3. Archive any customizations or extensions in the following directories:

```
[your Magento install dir]/app/code/local
[your Magento install dir]/app/code/community
```

4. Transfer the archives you just created to the development environment.

5. Copy `[your Magento install dir]/app/etc/local.xml` to your development environment.

## Restoring Configuration Settings

After you've exported the Magento database and extensions, you can revert the configuration changes you made as discussed in these sections:

- [Enabling Exception Logging](#)
- [Setting All Indexers for Update on Save \(EE only\)](#)

## Pre-Upgrade Tasks For Your Development Environment

In your development environment:

1. If necessary, create a Magento installation master directory; Magento installs in a `magento` subdirectory of that directory.

The Magento installation archive creates a `magento` subdirectory so if you want your Magento installation directory to be `/var/www/magento`, you don't need to do anything.



**Tip:** The name and path to the Magento installation directory in your development system does not have to be the same as on your production system, although it might make some tasks simpler.

2. Create a new database instance. For example, you can create a new MySQL database instance named `magento` using the following commands:

```
mysql -u root -p
```

Enter the remaining commands at the `mysql>` prompt

```
create database magento;
GRANT USAGE ON *.* TO magento@localhost IDENTIFIED BY 'magento';
GRANT ALL ON magento.* TO magento@localhost;
```

For MySQL versions later than 5.0.2 but earlier than 5.1.6, the following command is required:

```
GRANT SUPER ON *.* TO 'magento'@'localhost';
```

Exit the MySQL command shell:

```
exit
```

Verify the database exists using the following command:

```
mysql -u magento -p magento
```

If an error displays, repeat the commands. If the command succeeded, enter `exit` to return to the command prompt.

3. If you're using [Security Enhanced Linux \(SELinux\)](#) and Apache must initiate a connection to another host, you must run the commands discussed in this step.
  - a. To determine if SELinux is enabled, use the following command: `getenforce` `Enforcing` displays to confirm that SELinux is running.
  - b. Enter one of the following commands:  
CentOS:

```
setsebool -P httpd_can_network_connect=1
```

Ubuntu

```
setsebool -P apache2_can_network_connect=1
```

## Optional: Upgrading to PHP 5.4

PHP 5.3 is currently the latest PHP version available in the default repositories for Ubuntu and CentOS. PHP 5.3 works with CE 1.8, CE 1.9, EE 1.13, and EE 1.14.

We recommend PHP 5.4 for all of the preceding CE and EE versions because of the [new features and changes](#) in that release.

CE 1.8 and EE 1.13 both require a [PHP 5.4 patch](#). The patch is listed as [PHP 5.4 Compatibility](#) in the EE support portal.

### Upgrading to PHP 5.4—Ubuntu

Use the instructions on [phpave](#).

### Upgrading to PHP 5.4—CentOS

Enter the following commands in the order shown as a user with `root` privileges:

```
cd /tmp
rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
rpm -Uvh http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
yum --enablerepo=remi install httpd php php-common
```

The following resources are also available:

- [kb.parallels](#)
- [stackoverflow](#)

- remi repository

# Upgrading the File System, Solr, and Database

Upgrade your Magento installation by completing all of the following tasks in the order shown:

1. [Upgrade Requirements](#)
2. [Upgrading Assets on the Magento File System](#)
3. [Upgrading the Magento Database](#)
4. [Upgrading Solr \(\*EE 1.14.0.0 only\*\)](#)
5. [Finishing the File System](#)

## Upgrade Requirements

Before attempting your Magento upgrade, complete all the tasks discussed in:

- [Getting Ready for Your Upgrade](#)
- [Getting Magento CE or EE](#)

## Upgrading Assets on the Magento File System

This section discusses how to extract the Magento archive in your development system and manually copy over your customizations, themes, and extensions.

**!** **Important:** To avoid errors after upgrading, you *must* set up a parallel system. *Do not* upgrade your existing system or post-upgrade errors are likely to occur.

1. Make sure the CE or EE installation archive is located in the directory above where you want Magento to be installed.  
For example, to install Magento to `/var/www/magento`, copy the archive to `/var/www`.
2. Extract the archive. For example, `tar -zxf archive-name`  
This installs Magento CE or EE on the file system on your development system. Next, you'll overwrite some of these assets with customizations from your production system.
3. Extract the production system's `media` archive, overwriting the installation archive. You created the `media` archive as discussed in [Getting Ready For Your Upgrade](#).  
**Tip:** Copy the `media` archive to the same level in the directory structure to simplify the process. For example, if you created the `media` archive on your production system in your Magento install directory, the archive when extracted creates the `media` directory and all subdirectories. If you copy the `media` archive to the Magento installation directory on your development system, extracting it automatically replaces existing `media` subdirectory contents.
4. If necessary, move the `media` directory and subdirectories to `media` directory in your new Magento installation, overwriting the existing contents.
5. Extract the custom theme package archive you created as discussed in [Getting Ready For Your Upgrade](#).
6. If necessary, move the theme packages to the `[your Magento install dir]/app/design/frontend` and `[your Magento install dir]/skin/frontend` directories, as appropriate, overwriting the existing contents.
7. Extract the customization and extension archives you created as discussed in [Getting Ready For](#)

## Your Upgrade.

8. If necessary, move the extensions and customizations to the [your Magento install dir]/app/code/local and [your Magento install dir]/app/code/community directories, as appropriate, overwriting the existing contents.

## Updating the Magento Database

This section discusses how to:

- Import the production database data into the development database instance.
- In the development database instance, change the values of the paths `web/unsecure/base_url` and `web/secure/base_url` in the `core_config_data` table to the development server's IP address or hostname.

Import the production database data using your database manager tool.



**Important:** The database instance into which you import *must be empty*.

`mysql` syntax follows:

```
mysql -u root -p database-name < database-export-filename.sql
```

For example, if the database name is `magento` and the database export file name is `magento-db-export.sql`, enter

```
mysql -u root -p magento < magento-db-export.sql
```

The following sections discuss how to change the base URL paths from `http://prod.example.com` to `http://dev.example.com` using SQL commands and phpmyadmin:

- [Updating the Base URL the Database Using SQL Commands](#)
- [Updating the Base URL the Database Using phpmyadmin](#)

## Updating the Base URL the Database Using SQL Commands

This section discusses how to update the secure and unsecure URLs of your webstores and store views in the Magento database. The following example assumes you have two URLs, which every Magento installation has by default. Depending on your setup, you could have more (for additional store views, Content Delivery Networks (CDNs) and so on).

Use the following procedure to update *all* matching URLs in the database.

To update the database with the development system's base URLs using SQL commands:

1. Log in to the database server as any user.
2. Enter the following commands in the order shown:

```
mysql -u root -p
```

Enter the remaining commands at the `mysql>` prompt.

```
use magento-db-name;
SELECT * FROM core_config_data WHERE path LIKE '%base_url%';
```

3. In the resulting output, make note of the rows that include `web/unsecure/base_url` and `web/secure/base_url`.

In particular, you must know the value of the `config_id` for these rows.

4. Carefully examine the *entire URL* to avoid mistakes.
5. Enter the following command to update the values of `web/unsecure/base_url` and `web/secure/base_url`:

```
UPDATE core_config_data SET value='base-url' WHERE config_id=id1 or config_id=id2;
```

Example 1—`config_ids` are the same: if your base URL is `http://dev.example.com/`, the `config_id` of the row containing `web/unsecure/base_url` is 354 and the `config_id` of the row containing `web/secure/base_url` is 355, enter:

```
UPDATE core_config_data SET value='http://dev.example.com/' WHERE config_id=354 or config_id=355;
```

Example 2—`config_ids` are different: Same as preceding example except that `web/secure/base_url` uses `https://`

```
UPDATE core_config_data SET value='http://dev.example.com/' WHERE config_id=354;
UPDATE core_config_data SET value='https://dev.example.com/' WHERE config_id=355;
```

6. Verify the values are set correctly.

```
SELECT * FROM core_config_data WHERE path LIKE '%base_url%';
```

7. For other URLs, be careful to perform the replacement correctly.  
For example, store view base link URLs resemble the following:

```
http://prod.example.com/view1/
```

Make sure to update the URL like the following:

```
UPDATE core_config_data SET value='http://dev.example.com/view1/' WHERE config_id=377;
```

8. When you're done, run the following command again:

```
SELECT * FROM core_config_data WHERE path LIKE '%base_url%';
```

9. If the values are set correctly, enter `exit` at the `mysql>` prompt and continue with [Finishing the File System](#).

If not, repeat the tasks discussed in this section. *Do not continue* until the database is updated.

## Updating the Base URL the Database Using phpmyadmin

This section discusses how to use phpmyadmin to change the values of the paths `web/unsecure/base_url` and `web/secure/base_url` in the `core_config_data` table to the development server's IP address or hostname.

1. In a web browser's location or address field, enter `dev-web-server-host-or-ip/phpmyadmin`  
For example, if your web server address is `http://dev.example.com`, enter  
`http://dev.example.com/phpmyadmin`
2. Log in as the MySQL `root` user.
3. In the left navigation bar, click the name of your Magento database.

4. In the left navigation bar, click the **core\_config\_data** table.

5. At the top of the page, click the **Search** tab.

6. In the Search tab page, from the **Operator** list, click **LIKE**.

7. Enter **%base\_url%** in the **Value** field.

The following figure shows an example.

localhost ▶ magento ▶ core\_config\_data "Config Data"

Do a "query by example" (wildcard: "%")

| Field     | Type         | Collation       | Operator | Value      |
|-----------|--------------|-----------------|----------|------------|
| config_id | int(10)      |                 | =        |            |
| scope     | varchar(8)   | utf8_general_ci | LIKE     |            |
| scope_id  | int(11)      |                 | =        |            |
| path      | varchar(255) | utf8_general_ci | LIKE     | %base_url% |
| value     | text         | utf8_general_ci | LIKE     |            |

+ Options

Go

8. Click **Go**.

The following figure shows sample results.

|                          | config_id | Config Id | scope | Config Scope | scope_id | Config Scope Id                       | path | Config Path | value                            | Config Value |
|--------------------------|-----------|-----------|-------|--------------|----------|---------------------------------------|------|-------------|----------------------------------|--------------|
| <input type="checkbox"/> | 5         | default   |       |              | 0        | web/unsecure/base_url                 |      |             | http://prod.example.com/magento/ |              |
| <input type="checkbox"/> | 6         | default   |       |              | 0        | web/secure/base_url                   |      |             | http://prod.example.com/magento/ |              |
| <input type="checkbox"/> | 21        | default   |       |              | 0        | catalog/frontend/parse_url_directives | 1    |             |                                  |              |
| <input type="checkbox"/> | 46        | default   |       |              | 0        | catalog/seo/product_url_suffix        | html |             |                                  |              |
| <input type="checkbox"/> | 47        | default   |       |              | 0        | catalog/seo/category_url_suffix       | html |             |                                  |              |

Check All / Uncheck All With selected:

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

9. Click (Edit) in the row that includes the path `web/unsecure/base_url`.

10. Change the value to the hostname or IP address of your development server.

The following figure shows an example.

localhost ▶ magento ▶ core\_config\_data "Config Data"

Insert

| Field     | Type             | Function | Null | Value                                                        |
|-----------|------------------|----------|------|--------------------------------------------------------------|
| config_id | int(10) unsigned |          |      | 6                                                            |
| scope     | varchar(8)       |          |      | default                                                      |
| scope_id  | int(11)          |          |      | 0                                                            |
| path      | varchar(255)     |          |      | web/unsecure/base_url                                        |
| value     | text             |          |      | <input type="text" value="http://dev.example.com/magento/"/> |

Go

11. Click **Go**.

12. Repeat these tasks for the path that includes `web/secure/base_url`.

13. Exit phpmyadmin.

## Upgrading Solr (EE 1.14.0.0 only)

*This section applies to upgrading to EE 1.14.0.0 using the Solr search engine only.* If you're upgrading to a different version—or if you're not using Solr—skip this section and continue with [Finishing the File System](#).

Because of changes to the Solr schema, you must copy two files from Magento to your Solr installation. Failure to do so might prevent products from displaying on your web store. After copying the files, you must also reindex the catalog search index.

To update Solr:

1. If the Solr search engine is running, stop it.
2. Copy the following files from the Magento 1.14.0.0 installation to the equivalent directory on your Solr installation:

```
[your Magento install dir]/lib/Apache/Solr/conf/schema.xml
[your Magento install dir]/lib/Apache/Solr/conf/solrconfig.xml
```

For example, if Magento is installed in /var/www/html/magento and the example Solr configuration is installed in the /etc/solr/apache-solr-3.6.2/example/solr/conf directory on the same host, enter:

```
cp /var/www/html/magento/lib/Apache/Solr/conf/solrconfig.xml /etc/solr/apache-solr-3.6.2/example/solr/conf
cp /var/www/html/magento/lib/Apache/Solr/conf/schema.xml /etc/solr/apache-solr-3.6.2/example/solr/conf
```

3. Restart Solr.
4. After you copy the files, you must update the catalog search index. To update the catalog search index, open a command prompt window.
5. Change to the `shell` subdirectory of your Magento installation directory.

For example, on CentOS:

```
cd /var/www/html/magento/shell
```

6. Enter the following command:

```
php indexer.php --reindex catalogsearch_fulltext
```



**Note:** The preceding example showed paths used by the Solr example configuration. Solr ships with a sample configuration that Magento suggests you use for development only. Do not use the Solr example configuration in production.

## Finishing the File System

This section discusses how to edit copy `local.xml` and set file system permissions and ownership on the production system.

### Copying local.xml

1. Copy your production system's `local.xml` to [your Magento install dir]/app/etc.
2. Open `local.xml` in a text editor.
3. If necessary, change the database connection information in the `default_setup` element, as follows:

```
<default_setup>
```

```

<connection>
 <host><! [CDATA[change-if-necessary]]></host>
 <username><! [CDATA[change-if-necessary]]></username>
 <password><! [CDATA[change-if-necessary]]></password>
 <dbname><! [CDATA[change-if-necessary]]></dbname>
 <initStatements><! [CDATA[SET NAMES utf8]]></initStatements>
 <model><! [CDATA[mysql4]]></model>
 <type><! [CDATA[pdo_mysql]]></type>
 <pdoType><! [CDATA[]]></pdoType>
 <active>1</active>
</connection>
</default_setup>

```

4. Save your changes to `local.xml` and exit the text editor.

## Setting File and Directory Ownership and Privileges

Magento recommends the following ownership and privilege settings for files and directories in the Magento installation directory:

- The Magento installation directory and all subdirectories are owned by the web server user. This enables the web server to change files in these subdirectories but other users cannot access them (except a higher-level user such as `root`).
- All directories have 700 permissions (`drwx-----`). 700 permissions give full control (that is, read/write/execute) to the owner and no permissions to anyone else.
- All files have 600 permissions (`-rw-----`). 600 permissions mean the owner can read and write but other users have no permissions.

**Note:** The way you set permissions and ownership depends on whether Magento is running on a dedicated or hosted system:

- **Hosted:** A Magento server located on a hosting provider. A `hosted` system typically does not enable you to elevate to `root`. The web server typically runs as an ordinary user. Magento assumes you log in as this user to start and stop the web server and that you already own all the files and directories in the Magento installation directory. You can use `chmod` to change permissions on files and directories.
- **Dedicated:** A Magento server you control and operate. Unlike a hosted system, you can elevate to `root` and, as `root`, you can use the `chown` and `chmod` commands to set ownership and privileges in the Magento installation directory.

To set privileges and ownership:

1. Log in to your Magento server.

2. Change to your Magento installation directory:

```
#Ubuntu example
cd /var/www/magento

#CentOS example
cd /var/www/html/magento
```

3. **Dedicated Magento server only.** Enter the following command to set ownership of the Magento installation directory and all its subdirectories:

```
chown -R web-server-user-name .
```

For example, on Ubuntu where Apache usually runs as `www-data`, enter

```
chown -R www-data .
```

4. Enter the following commands to set directory permissions to 700 and file permissions to 600:

```
find . -type d -exec chmod 700 {} +
find . -type f -exec chmod 600 {} +
```

## Running the URL Redirect Script (EE 1.13.0.0 and 1.13.0.1 only)

If you're upgrading to EE 1.14 from EE 1.13.0.0 or 1.13.0.1, see [EE 1.13.1.0 Upgrade: Running the EE 1.13.0.0 or EE 1.13.0.1 URL Redirect Script](#) now. When you're done, return here to continue your upgrade.

## Completing the Magento Upgrade

To complete the upgrade, go to your Magento base URL in a web browser. The first time you go to your Magento base URL, server-side scripts run to update the database. Depending on the amount of data in your database, this process can take a long time.



**Important:** *Completing the upgrade requires downtime on your production system.*

If you continue to take orders on your production system, there is no way to reconcile all of the new orders in the replacement development system. You can reduce the amount of downtime to a minimum by thoroughly testing your development system before shutting down production.

If you're upgrading from a version earlier than CE 1.4 or EE 1.7, Magento *strongly recommends* the two-step upgrade approach discussed in [Upgrade Path](#). In addition, you should expect the upgrade and testing process to take a long time and to expect more downtime for your production system.

Complete the following tasks in the order shown:

1. [Running the Upgrade Scripts and Fixing Exceptions](#)
2. [Running the URL Redirect Script \(EE 1.12.x and earlier only\)](#)
3. [Post-Upgrade Tasks \(EE 1.13.0.1 Only\)](#)
4. To apply the [security patch](#) released in October, 2016, see [How to Apply the SUPEE-8788 Patch](#).
5. [Setting Up the Magento Cron Job](#)
6. [Clearing Magento var/ Subdirectories](#)
7. [Verifying the Status of Indexers](#)
8. [Testing the Magento Upgrade](#)
9. [Taking Your Old Production System Offline](#)

## Running the Upgrade Scripts and Fixing Exceptions

The first step in your upgrade is to run server-side upgrade scripts. Depending on the nature of your customizations and extensions, and how many customers and products are in your database, these scripts can take a long time to run and can result in exceptions. You must resolve all exceptions before continuing to the next step in the process.

This step in the upgrade process is iterative; that is, you'll probably run through it more than once.



**Note:** Your production system remains up during this step in the upgrade process. You can continue taking and shipping orders and also making changes in the Admin Panel.

To run the upgrade scripts:

1. In a web browser address or location field, enter your Magento installation's base URL.



**Note:** If you upgraded Magento on the same server, you might need to update your Apache web server's virtual host configuration to reference the new Magento installation directory. For more information about configuring virtual hosts, see the [Apache Virtual Host documentation](#).

(You updated the Magento database with the development system base URL as discussed in [Updating the Magento Database](#).)

2. Wait while upgrade scripts run on the server.
3. If fatal errors or exceptions display in the browser, they are most likely related to customizations.  
To resolve fatal errors, you must:
  - a. Analyze the errors to determine the root cause.  
You can find errors in the web server's error log or in `[your Magento install dir]/var/log`.
  - b. Fix the cause in your extension code.
  - c. In your development environment, drop the Magento database.
  - d. Re-create the development database and import production data into it as discussed in [Updating the Magento Database](#).
  - e. Use the following command to delete the contents of the Magento var/ subdirectories:

```
rm -rf [your Magento install dir]/var/cache [your Magento install dir]/var/full_page_cache \
[your Magento install dir]/var/locks [your Magento install dir]/var/session
```



**Note:** Some of these directories might be empty; this is normal.

- f. Clean up the file system as discussed in [Finishing the File System](#).
- g. Repeat step 1 in this section.
- h. Repeat the tasks discussed in this section until no more fatal errors display in the browser.

## Post-Upgrade Tasks (EE 1.13.0.1 Only)

*This section applies to upgrading from 1.13.0.1 only.* If you're upgrading CE or if you're upgrading from a different EE version, continue with [Setting Up the Magento Cron Job](#).

After successfully running the Magento upgrade and fixing errors, enter the following command from the Magento root directory:

```
php -f shell/indexer.php -- --reindexall
```

This command runs a full reindex and it might take a long time, depending on the size of your database.

After the reindex completes, enter the following command from your Magento installation directory:

```
rm -rf var/cache var/full_page_cache var/locks
```

Continue with [Setting Up Magento Cron Jobs](#).

## Running the URL Redirect Script (EE 1.12.x and Earlier Only)

If you're upgrading to EE 1.14 from EE 1.12 or earlier, see [EE 1.14 Upgrade: Running the EE 1.12 and Earlier URL Redirect Script](#) now. When you're done, return here to continue your upgrade.

## Setting Up the Magento Cron Job

You should now set up your Magento cron job as discussed [here](#).

## Clearing Magento var/ Subdirectories

Run the following command to clean the Magento var/ subdirectories:

```
rm -rf [your Magento install dir]/var/cache [your Magento install dir]/var/full_page_cache [your Magento install dir]/var/locks
```



**Note:** Some of these directories might be empty; this is normal.

## Verifying the Status of Indexers

After the cron job runs, all indexers should be in a Ready status. Before testing your upgrade, verify all indexers are Ready; otherwise, your testing will be inconclusive.

To verify the status of indexers:

1. Log in to the Admin Panel as an administrator.
2. Click **System > Index Management**.

Verify that all indexers have a status of Ready, as the following figure shows.

| Index Management         |                                |                                                          |                |        |                 |                         |
|--------------------------|--------------------------------|----------------------------------------------------------|----------------|--------|-----------------|-------------------------|
|                          |                                | Description                                              | Mode           | Status | Update Required | Updated At              |
| <input type="checkbox"/> | Product Attributes             | Index product attributes for layered navigation building | Update on Save | READY  | NO              | Jul 19, 2013 5:40:36 AM |
| <input type="checkbox"/> | Tag Aggregation Data           | Rebuild Tag aggregation data                             | Update on Save | READY  | NO              | Jul 19, 2013 5:40:36 AM |
| <input type="checkbox"/> | URL Redirects                  | Index URL rewrites                                       |                | READY  |                 | Never                   |
| <input type="checkbox"/> | Stock Status                   | Index Product Stock Status                               |                | READY  |                 | Never                   |
| <input type="checkbox"/> | Catalog Search Index           | Rebuild Catalog product fulltext search index            |                | READY  |                 | Never                   |
| <input type="checkbox"/> | Product URL Rewrites           | Index products URL rewrites                              |                | READY  |                 | Never                   |
| <input type="checkbox"/> | Category URL Rewrites          | Index categories URL rewrites                            |                | READY  |                 | Never                   |
| <input type="checkbox"/> | Catalog Category/Product Index | Indexed category/products association                    |                | READY  |                 | Never                   |
| <input type="checkbox"/> | Catalog product price          | Index product prices                                     |                | READY  |                 | Never                   |

3. If indexers are *not* Ready, use the following guidelines:

| INDEXER STATUS | SUGGESTED ACTION                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scheduled      | Wait for cron to finish or run <a href="http://magento-host-name/cron.php">http://magento-host-name/cron.php</a> from a web browser.                                                                                                                                                                                                                                                                                       |
| Processing     | Wait for cron to finish or run <a href="http://www.example.com/magento/cron.php">cron.php</a> from a web browser. (For example, <a href="http://www.example.com/magento/cron.php">http://www.example.com/magento/cron.php</a> )<br><br>If indexers are Processing for an extended period of time, run the command discussed in <a href="#">Clearing Magento var/ Subdirectories</a> and refresh the Index Management page. |

## Testing the Magento Upgrade

After you have upgraded with no fatal errors or exceptions, Magento strongly recommends you thoroughly test the upgrade in your development environment as follows:

1. Test all extensions and customizations thoroughly.  
Make sure the UIs perform as expected, make sure the extensions behave properly, and so on.
2. Run some orders, making sure prices are calculated correctly and that orders go through the configured payment mechanisms.
3. If you encounter issues, [get help](#).
4. Before making the development system live, back up everything in your production and development environments again.  
The way you make your development system live is up to you; a simple way is to update your DNS server to point to the development IP address.

## Taking Your Old Production System Offline

The last step in upgrading is to take your production system offline and switching to your development system—which then becomes the production system from that point on.



**Important:** Before continuing, make sure all errors have been resolved and that you have thoroughly tested the upgrade as discussed in the preceding section.

To switch from your production to development system:

1. Put your production system in maintenance mode so it cannot accept orders or other changes.  
To do so, create an empty file named `[your Magento install dir]/maintenance.flag`.
2. Test it by going to your Magento web store's front page.  
The following message displays if you set up maintenance mode correctly: **The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.**
3. If necessary, update the development database again to get any final transactions processed by your current production system.  
For details, see [Updating the Magento Database](#).
4. Put your production system in maintenance mode so it cannot process orders.
5. Make your development system live.  
The way you do this is up to you; typically, you can configure your DNS server to send requests for your web store domain to the development system's IP address. Be aware that propagating the change to other DNS servers can take some time.
6. Verify your former development system is functioning as your new Magento web store in production.
7. Shut down the old production system.

Congratulations! You successfully upgraded! Review our [welcome page](#) to see the improvements you're getting.

## Setting Magento File System Permissions and Ownership After Upgrade

To secure your Magento installation after the upgrade, see [After You Install Magento: Recommended File System Ownership and Privileges](#).

# Troubleshooting

There is a known issue after upgrading to EE 1.13.1 that affects you *only* if you do *not* follow the recommended procedure to upgrade to a new environment as discussed in [Getting Ready For Your Upgrade](#).

**Symptom:** After completing the upgrade, when you log in to the Admin Panel and click **System > Configuration**, a fatal error similar to the following displays in your browser:

```
Class 'Mage_Googlecheckout_Helper_Data' not found in /var/www/html/magento/app/Mage.php on line 547
```

## Solution:

1. Close the Admin Panel browser window.
2. As a user with **root** privileges, delete all files *except config.xml* from the following directory:  
[your Magento install dir]/app/code/core/Mage/GoogleCheckout/etc
3. When you log back in to the Admin Panel, everything works as expected.

In the event of any other issue during the upgrade, see [Getting Help With Your Installation or Upgrade](#).

22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Installing and Verifying Magento Community Edition (CE) and Enterprise Edition (EE)

Edit this page on GitHub 

## Contents

- [Overview](#)
- [Prerequisites](#)
- [Preparing Your Server for Magento CE or EE](#)
- [Installing Magento CE or EE](#)
- [Verifying that Magento CE or Magento EE Installed Successfully](#)
- [Setting Up Cronjobs](#)
- [Setting Magento File System Permissions and Ownership After Installation](#)
- [Troubleshooting](#)

## Overview

This article discusses how to install:

- Magento Community Edition (CE) 1.8.x or 1.9.x
- Magento Enterprise Edition (EE) 1.13.x or 1.14.x

To install Magento CE or Magento EE, use the following high-level guidelines:

1. Set up a system that meets the [Magento system requirements](#).
2. Install prerequisite software (web server, PHP, and required PHP extensions).
3. Set up a MySQL Magento database instance.
4. Run the Magento installer to complete the installation.
5. Verify that Magento installed correctly.



**Important:** Perform *all installations* to Magento the latest version of CE 1.9 or Magento EE 1.14 to get the latest fixes, features, and security updates.

## Prerequisites

Before starting your installation:

- Get the Magento installation archive as discussed in [Getting Magento CE or EE](#).
- Set up one or more hosts that meet the [Magento system requirements](#).
- If you are setting up more than one web node with load balancing, set up and test that part of your system *before* you install Magento.
- If your database server and web server are located on different hosts, [get help](#) before proceeding.
- Make sure you can back up your entire system at various points during the installation so you can roll back in the event of issues.

Magento *strongly* recommends you observe the following guidelines when you set up your Magento database:

- Magento for the first time uses [MySQL database triggers](#) to improve database access during reindexing. Magento does not support any custom triggers in the Magento database because custom triggers can introduce incompatibilities with future Magento versions.
- Familiarize yourself with [these potential MySQL trigger limitations](#) before you continue.
- If you use MySQL database replication, be aware that Magento *does not* support MySQL statement-based replication. Make sure you use only [row-based replication](#).

More information, including software prerequisites, can be found in the next section.

## Preparing Your Server for Magento CE or EE

The following sections discuss how to download and install prerequisite software and install Magento CE or EE on a system running:

- Ubuntu 10 or later, or CentOS 6
- Apache 2.x
- nginx 1.7.x
- PHP 5.4 and required extensions
- MySQL database

### Notes:

- Discussing how to configure Secure Sockets Layer (SSL) is beyond the scope of this guide. You can consult some general resources such as [Apache](#), [zytrax.com](#), or the [OpenSSL documentation page](#).
- Discussing how to [install and configure nginx](#) is beyond the scope of this article.
- Review the MySQL developer site's recommendations for [MySQL password security](#).
- Before you continue, make sure the system on which you install Magento meets the [Magento system requirements](#).

Complete the tasks discussed in the following sections in the order shown:

1. Installing and configuring required prerequisite Software for [Ubuntu](#) or [CentOS](#)
2. [SELinux Prerequisite](#)
3. [Opening Ports In Your Firewall](#)
4. [Installing Optional Software](#)
5. [Creating a Magento Database Instance](#)
6. [Extracting the Magento Installation Archive](#)
7. [Installing Optional Sample Data](#)



**Note:** Unless otherwise noted, all tasks discussed in this article must be performed by a user with `root` privileges.

## SELinux Prerequisite

[Security Enhanced Linux \(SELinux\)](#) enables CentOS and Ubuntu administrators greater access control over their servers. If you're using SELinux *and* Apache must initiate a connection to another host, you must run the commands discussed in this section.

If Apache and the database server are on the same host, you can skip this section and continue with [Opening Ports In Your Firewall](#).

To enable Apache to initiate a connection to another host with SELinux enabled:

1. To determine if SELinux is enabled, use the following command:

```
getenforce
```

`Enforcing` displays to confirm that SELinux is running.

2. Enter one of the following commands:

- CentOS

```
setsebool -P httpd_can_network_connect=1
```

- Ubuntu

```
setsebool -P apache2_can_network_connect=1
```

## Opening Ports In Your Firewall

Depending on your security requirements, you might find it necessary to open port 80 and other ports in your firewall. Because of the sensitive nature of networking security, Magento strongly recommends you consult with your IT department before proceeding. Following are some suggested references:

- Ubuntu: [Ubuntu documentation page](#).
- CentOS: [CentOS how-to](#).

## Installing Optional Software

This section discusses how to create or install the following:

- Install and configure Network Time Protocol (NTP) which enables you to synchronize the system clock with pool servers located around the world. NTP is particularly useful for synchronizing the clocks of multiple hosts.
- Create a [phpinfo.php](#) file
- Install and configure the [phpmyadmin](#) database administration utility.

These optional utilities might assist you with your installation; however, they are not required to install or use Magento.

To skip installing these utilities, go directly to [Creating a Magento Database Instance](#).

See the following sections:

- [Installing and Configuring Network Time Protocol \(NTP\)](#)
- [Creating phpinfo.php](#)
- [Installing phpmyadmin](#)

### Installing and Configuring Network Time Protocol (NTP)

NTP enables servers to synchronize their system clocks using [globally available pool servers](#). Magento recommends you use NTP servers you trust, whether they are dedicated hardware solutions your internal network or external, public servers.

If you are deploying Magento on multiple hosts, NTP is a simple way to guarantee their clocks are all synchronized, no matter what time zone the servers are in.

To install and configure NTP:

- CentOS only.*
  - Enter the following command to find the appropriate NTP software: `yum search ntp`
  - Select a package to install. For example, `ntp.x86_64`.
  - Install the package. `yum -y install ntp.x86_64`
- Ubuntu only.* Enter the following command to install NTP: `apt-get install ntp`
- Select the NTP pool servers you wish to use.  
Selecting pool servers is up to you. If you use NTP pool servers, ntp.org recommends you use [pool servers](#) that are close to your servers' time zone as discussed on the [NTP pool project help page](#). If you have a private NTP server that is available to all hosts in your Magento deployment, you can use that server instead.
- Open `/etc/ntp.conf` in a text editor.
- Look for lines similar to the following:

```
server 0.centos.pool.ntp.org
server 1.centos.pool.ntp.org
server 2.centos.pool.ntp.org
```

- Replace those lines or add additional lines that specify your NTP pool server or other NTP servers. It's a good idea to specify more than one.

An example of using three United States-based NTP servers follows:

```
server 0.us.pool.ntp.org
server 1.us.pool.ntp.org
server 2.us.pool.ntp.org
```

- Save your changes to `/etc/ntp.conf` and exit the text editor.

- CentOS only.* Enter the following command so that NTP starts when the server starts.

```
chkconfig ntpd on
```

9. Restart the service.

- CentOS

```
service ntpd restart
```

- Ubuntu

```
service ntp restart
```

10. Enter the `date` command to check the server's date.

If the date is incorrect, make sure the NTP client port (typically, UDP 123) is open in your firewall. Try the `ntpdate pool-server-host-name` command. If it fails, search for the error it returns.

If all else fails, try restarting the server.

## Creating phpinfo.php

[phpinfo.php](#) displays a large amount of information about PHP and its extensions. Add the following code anywhere in your web server's docroot:

```
<?php
// Show all information, defaults to INFO_ALL
phpinfo();
```

For more information, see the [phpinfo manual page](#).

To view the results, enter the following URL in your browser's location or address field:

```
http://web-server-ip-or-host/phpinfo.php
```

If a 404 (Not Found) error displays, check the following:

- Start the web server if necessary.
- Make sure your firewall allows traffic on port 80.  
[Help for Ubuntu](#)  
[Help for CentOS](#)

## Installing phpmyadmin

[phpmyadmin](#) is an easy-to-use, free database administration utility. You can use it to check and manipulate the contents of your database. You must log in to [phpmyadmin](#) as the MySQL database administrative user.

For more information about [phpmyadmin](#), see the [phpmyadmin home page](#).

For more detailed information about installation, see the [phpmyadmin installation documentation](#).

To install [phpmyadmin](#) on CentOS:

1. Download the epel RPM for the version of CentOS you're using. A sample follows.

```
cd /tmp
wget http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
rpm -ivh epel-release-6-8.noarch.rpm
```

2. Install [phpmyadmin](#) as follows:

```
yum -y install phpmyadmin
```

3. Authorize access to [phpmyadmin](#) from your machine's IP address.

- a. Open the following file for editing:

```
vim /etc/httpd/conf.d/phpMyAdmin.conf
```

- b. Replace the following IP address with your IP address

```
#Require ip 127.0.0.1
```

For example,

```
Require ip 192.51.100.101
```

- c. Replace the following IP with your IP address

```
#Allow from 127.0.0.1
```

For example,

```
Allow from 192.51.100.101
```

4. Save your changes to `/etc/httpd/conf.d/phpMyAdmin.conf` and exit the text editor.

5. Restart Apache.

```
service httpd restart
```

6. To use phpmyadmin, enter the following command in your browser's address or location field:

```
http://host-or-ip-address/phpmyadmin
```

7. When prompted, log in using your MySQL database `root` or administrative user's username and password.

To install phpmyadmin on Ubuntu:

1. Use the following command: `apt-get install phpmyadmin`

2. Follow the prompts on your screen to complete the installation.

3. To use phpmyadmin, enter the following URL in your browser's address or location field: `http://host-or-ip-address/phpmyadmin`

4. When prompted, log in using your MySQL database `root` or administrative user's username and password.

## Creating a Magento Database Instance

This section discusses how to create a new database instance for Magento. Although a new database instance is recommended, you can optionally install Magento into an existing database instance. If you choose to do that, skip this section and continue with [Installing Optional Sample Data](#).



**Note:** Before you continue, review the information about MySQL discussed in [Prerequisites](#).

To create a new database instance:

1. Log in to your database server as any user.

2. Enter the following commands in the order shown to create a database instance named `magento`:

```
mysql -u root -p
#Enter the remaining commands at the mysql> prompt.

create database magento;
GRANT ALL ON magento.* TO magento@localhost IDENTIFIED BY 'magento';
```

3. For MySQL versions between 5.0.2 and 5.1.6, you must enter this command:

```
GRANT SUPER ON *.* TO 'magento'@'localhost';
```

4. After you're done, enter `exit`

5. Test the database instance.

```
mysql -u magento -p
```

Messages similar to the following display to confirm you successfully created the database instance. If errors display, repeat the preceding commands.

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 5.1.67 Source distribution

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

## Extracting the Magento Installation Archive

Extract the Magento installation archive on your web server's docroot.

The path on Ubuntu is typically `/var/www`

The path on CentOS is typically `/var/www/html`

Examples follow:

- Ubuntu

```
cd /var/www
cp /path/magento-install-archive .
tar -zxf magento-install-archive-name
```

- CentOS

```
cd /var/www/html
cp /path/magento-install-archive-name .
tar -zxf magento-install-archive
```

To optionally install Magento sample data, continue with the next section.

To install Magento without sample data, continue with [Setting File and Directory Ownership and Privileges](#).

## Installing Optional Sample Data

Magento provides sample data you can optionally install to give you an idea of what products look like in the Admin Panel and in your web store. You can use this sample data to learn how Magento works and to experiment with custom themes.

You must install sample data on the file system and in the database *before* you install Magento.



**Note:** Only if your database is shared between more than one Magento installation. Database table names used by the sample data are not configurable. To use sample data with a new Magento installation, you must manually add a prefix to all sample data tables and use the same prefix when you install Magento.

To skip sample data installation, continue with [Setting File and Directory Ownership and Privileges](#).

To install Magento sample data:

1. Download the sample data from one of the locations discussed in [Getting Magento CE or EE](#).
2. If necessary, transfer the sample data to your Magento server.
3. On your Magento server, extract the sample data archive to a directory other than your Magento installation directory.
4. Make note of the path to the sample data's `media` subdirectory.
5. Change to the `[your Magento install dir]/media` directory.

```
#Ubuntu example
cd /var/www/magento/media
```

```
#CentOS example
cd /var/www/html/magento/media
```

6. As a user with privileges to write to the Magento installation directories (typically the web server user), copy the sample data's `media` directory and subdirectories to your Magento installation directory.

For example, to copy Magento sample data from `/home/username` to `/var/www/magento`, enter

```
cp -R /home/username/media/* .
```

7. *Magento CE 1.9 and Magento EE 1.14 only.* You must also copy the sample data's `skin` directory to `[your Magento install dir]/skin` as follows:

For example, to copy Magento skin files from `/home/username/skin` to `/var/www/magento/skin`, enter

```
cd [your Magento install dir]/skin
cp -R /home/username/skin/* .
```

8. Import the CE or EE sample data into your MySQL database as follows:

```
mysql -u root -p magento-db-instance-name < path-to-sample-data-extract-dir/sample-data-filename.sql
```

EE 1.14 example

```
mysql -u root -p magento < /home/username/magento_sample_data_for_1.14.0.0.sql
```

## Setting File and Directory Ownership and Privileges

Magento recommends the following ownership and privilege settings for files and directories in the Magento installation directory:

- The Magento installation directory and all subdirectories are owned by the web server user.

This enables the web server to change files in these subdirectories but other users cannot access them (except a higher-level user

such as root).

- All directories have 700 permissions (drwx-----).  
700 permissions give full control (that is, read/write/execute) to the owner and no permissions to anyone else.
- All files have 600 permissions (-rw-----).  
600 permissions mean the owner can read and write but other users have no permissions.

**Note:** The way you set permissions and ownership depends on whether Magento is running on a dedicated or hosted system:

- *Hosted*: A Magento server located on a hosting provider. A *hosted* system typically does not enable you to elevate to `root`. The web server typically runs as an ordinary user. Magento assumes you log in as this user to start and stop the web server and that you already own all the files and directories in the Magento installation directory. You can use `chmod` to change permissions on files and directories.
- *Dedicated*: A Magento server you control and operate. Unlike a hosted system, you can elevate to `root` and, as `root`, you can use the `chown` and `chmod` commands to set ownership and privileges in the Magento installation directory.

To set privileges and ownership:

1. Log in to your Magento server.
2. Change to your Magento installation directory:

```
#Ubuntu example
cd /var/www/magento

#CentOS example
cd /var/www/html/magento
```

3. *Dedicated Magento server only*. Enter the following command to set ownership of the Magento installation directory and all its subdirectories:

```
chown -R web-server-user-name .
```

For example, on Ubuntu where Apache usually runs as `www-data`, enter

```
chown -R www-data .
```

4. Enter the following commands to set directory permissions to 700 and file permissions to 600:

```
find . -type d -exec chmod 700 {} +
find . -type f -exec chmod 600 {} +
```

## Installing Magento CE or EE

This section discusses how to run the Magento installer, using Magento EE as an example. The Magento CE installer has exactly the same options; only the appearance is different.

**!** **Important:** The procedure that follows assumes that your web server and database server are on the same host. If they are installed on different hosts, additional tasks are required. [Get help](#) before you continue your installation.

To install Magento CE or EE:

1. Complete all of the tasks discussed earlier in this article.
2. Enter the following URL in your web browser's address or location field:

```
web-server-ip-or-host:port/magento-path/magento
```

For example, if your web server is `http://www.example.com`, listens on port 80, and Magento is installed in the web server docroot's `magento` subdirectory, enter

```
http://www.example.com/magento
```

The following page displays.



|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Installation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <p style="color: red; font-weight: bold;">Welcome to Magento's Installation Wizard!</p> <p><a href="#">Download</a><br/> <a href="#">License Agreement</a><br/> <a href="#">Localization</a><br/> <a href="#">Configuration</a><br/> <a href="#">Create Admin Account</a><br/> <a href="#">You're All Set!</a></p> <p>Having trouble installing Magento? Check out our <a href="#">Installation Guide</a></p> |
| <div style="border: 1px solid #ccc; padding: 10px;"> <p style="text-align: center;"><b>MAGENTO™ ENTERPRISE EDITION<br/>END USER LICENSE AGREEMENT</b></p> <p>This End User License Agreement ("Agreement") is entered into by and between X.commerce, Inc. through its Magento division ("Magento"), and the Licensee executing the Magento Order Form (defined below). The parties agree as follows:</p> <p><b>TERMS AND CONDITIONS</b></p> <p><b>1. License</b></p> <p>1.1. Subject to Licensee's payment of the applicable fees and to Licensee's compliance with other terms and conditions of this Agreement, Magento grants Licensee a non-transferable, non-assignable, non-sublicensable, worldwide license to copy the Software for the purpose of installing and using it on a computer and solely for internal purposes, in accordance with the Software's technical documentation and solely during the periods and on the maximum number of Designated Servers specified in one or more applicable Magento or Magento-authorized reseller ordering schedules (the "Magento Order Form") executed with Licensee.</p> <p><input checked="" type="checkbox"/> I agree to the above terms and conditions.</p> <p style="text-align: right;"><a href="#">Continue</a></p> </div> |                                                                                                                                                                                                                                                                                                                                                                                                               |

3. Select the checkbox that indicates you agree to the terms and conditions, and click **Continue**.
4. On the Localization page, enter the following information.

| OPTION           | MEANING                                                                                               |
|------------------|-------------------------------------------------------------------------------------------------------|
| Locale           | From the list, click the option that best matches the locale in which your Magento server resides.    |
| Time Zone        | From the list, click the option that best matches the time zone in which your Magento server resides. |
| Default currency | From the list, click the default currency to use on your Magento web store.                           |

The following figure shows an example of configuring Magento for US English in the US Central time zone and using the US Dollar as the default currency.

|                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Installation                                                                                                                                                                                                                                                                                                                              | <b>Localization</b>                                                                                                                                                                                                                                                                                                                                                       |
| <p><a href="#">Download</a><br/> <a href="#">License Agreement</a><br/> <a href="#">Localization</a> <b>Localization</b><br/> <a href="#">Configuration</a><br/> <a href="#">Create Admin Account</a><br/> <a href="#">You're All Set!</a></p> <p>Having trouble installing Magento? Check out our <a href="#">Installation Guide</a></p> | <div style="border: 1px solid #ccc; padding: 10px;"> <p><b>Locale settings</b></p> <p>Locale * English (United States) / English (United States)</p> <p>Time Zone * Central Standard Time (America/Chicago)</p> <p>Default Currency * US Dollar</p> <p style="text-align: right;">* Required Fields</p> <p style="text-align: right;"><a href="#">Continue</a></p> </div> |

5. Click **Continue**.
- The Configuration page displays. Each of its three sections is discussed in the following step.
6. In the Configuration page, enter the following information:
  - a. In the Database Configuration section, enter the following information.

| OPTION        | MEANING                                                                                                                                                                                                                |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Database Type | From the list, click <b>MySQL</b> .                                                                                                                                                                                    |
| Host          | Enter the database server's fully qualified hostname or IP address. Use the default setting of <b>localhost</b> if your database server is on the same host as your web server.                                        |
| Database Name | Enter the name of the Magento database instance in which you want to install the Magento database tables.                                                                                                              |
| User Name     | Enter the username of the Magento database instance owner.                                                                                                                                                             |
| User Password | Enter the Magento database owner's password.                                                                                                                                                                           |
| Tables Prefix | (Optional.) Use only if you're installing the Magento database tables in a database instance that has Magento tables in it already. In that case, enter a prefix to identify the Magento tables for this installation. |
|               | Some customers have more than one Magento instance running on a server with all tables in the same database. This option enables those                                                                                 |

customers to share the database server with more than one Magento installation.

A sample follows.

**Database Connection**

**Database Type**: MySQL

**Host \***: localhost

**Database Name \***: magento

You can specify server port, ex.: localhost:3307  
If you are not using default UNIX socket, you can specify it here instead of host, ex.: /var/run/mysqld/mysqld.sock

**User Name \***: magento

**User Password**: \*\*\*\*\*

**Tables Prefix**:

(Optional. Leave blank for no prefix)

b. In the Web Access Options section, enter the following information.

| OPTION                                        | MEANING                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Base URL                                      | Enter the base URL to use to access the Magento Admin Panel and your Magento web store.                                                                                                                                                                                                                                                       |
| Admin Path                                    | Enter the path to access the Admin Panel. This path is appended to <b>Base URL</b> .<br>For example, if <b>Base URL</b> is <code>http://www.example.com</code> and <b>Admin Path</b> is <code>admin</code> , the Admin Panel's URL is <code>http://www.example.com/admin</code> —provided you configured your web server for server rewrites. |
| Enable Charts                                 | Select the checkbox to display charts on the Admin Panel.                                                                                                                                                                                                                                                                                     |
| Skip Base URL Validation Before the Next Step | Clearing this checkbox validates your server's base URL by performing an HTTP GET. Clear this checkbox unless your web server's base URL is not verifiable; for example, in a development or test environment.                                                                                                                                |
| Use Web Server (Apache) Rewrites              | Select this checkbox to enable the use of the Apache <code>mod_rewrite</code> module. You can select this checkbox only if you configured Apache to use server rewrites.                                                                                                                                                                      |
| Use Secure URLs (SSL)                         | Select this checkbox only if your web server supports SSL.                                                                                                                                                                                                                                                                                    |

A sample follows.

**Web access options**

**Base URL \***: http://10.249.149.45/magento/

**Admin Path \***: admin

Additional path added after Base URL to access your Administrative Panel (e.g. admin, backend, control etc.).

**Enable Charts**  
Enable this option if you want the charts to be displayed on Dashboard.

**Skip Base URL Validation Before the Next Step**  
Check this box only if it is not possible to automatically validate the Base URL.

**Use Web Server (Apache) Rewrites**  
You could enable this option to use web server rewrites functionality for improved search engines optimization.  
Please make sure that `mod_rewrite` is enabled in Apache configuration.

**Use Secure URLs (SSL)**  
Enable this option only if you have SSL available.

c. In the Session Storage Options section, click one of the following options:

- a. • **File** to store user session data on the file system in the [your Magento install dir]/var/session directory.  
File-based session storage is appropriate unless the Magento file system access is slow or you have a clustered database.
  - b. • **Database** to store user session data in the database.  
Choose database storage if you have a clustered database; otherwise, there might not be much benefit over file-based storage.
- d. Click **Continue**.

**Important:** If errors display, you must resolve them before continuing.

e. In the Create Admin Account page, enter the following information.

| OPTION | MEANING |
|--------|---------|
|        |         |

|                  |                                                                                                                                                                                                                        |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| First Name       | Enter the first name of the user you want to identify as the Magento web store owner.                                                                                                                                  |
| Last Name        | Enter the user's last name.                                                                                                                                                                                            |
| Email            | Enter the user's email address.                                                                                                                                                                                        |
| Username         | Enter the Magento Admin Panel administrator's username. (You can create additional Magento administrators later.)                                                                                                      |
| Password         | Enter the user's password.                                                                                                                                                                                             |
| Confirm Password | Enter the user's password again for verification.                                                                                                                                                                      |
| Encryption Key   | If you have one, enter a key to encrypt sensitive data in the Magento database. If you don't have one, Magento generates one for you.<br>The encryption key is stored in [your Magento install dir]/app/etc/local.xml. |

A sample follows.

### Create Admin Account

Personal Information

|                                                      |                                   |
|------------------------------------------------------|-----------------------------------|
| First Name *                                         | Last Name *                       |
| <input type="text" value="Magento"/>                 | <input type="text" value="User"/> |
| Email *                                              |                                   |
| <input type="text" value="magentouser@example.com"/> |                                   |

Login Information

|                                        |                                    |
|----------------------------------------|------------------------------------|
| Username *                             | Confirm Password *                 |
| <input type="text" value="admin"/>     | <input type="text" value="*****"/> |
| Password *                             |                                    |
| <input type="password" value="*****"/> |                                    |

Encryption Key

Magento uses this key to encrypt passwords, credit cards and more. If this field is left empty the system will create an encryption key for you and will display it on the next page.

\* Required Fields

[Continue](#)

- f. Click **Continue**.

The following page displays to indicate a successful installation.



|                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Installation</b><br><br><a href="#">Download</a><br><a href="#">License Agreement</a><br><a href="#">Localization</a><br><a href="#">Configuration</a><br><a href="#">Create Admin Account</a><br><b>You're All Set!</b> | <b>You're All Set!</b><br><br>Get ready to experience Open-Source eCommerce Evolved.<br>Before you continue to your store, please make a note of your encryption key (Magento uses it to encrypt passwords, credit cards and more).<br><b>c39ef785663366602978a5G47bc5929</b><br>(Make sure you keep it in a safe place.) |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Having trouble installing Magento? Check out our [Installation Guide](#)

[Go to Frontend](#) [Go to Backend](#)

## Verifying that Magento CE or Magento EE Installed Successfully

To make sure Magento installed properly, you should log in to the Admin Panel to verify the Magento version.

You can also import products into Magento or perform other tasks that verify you can write to the database.

- a. In a web browser's location or address field, enter the URL to the Admin Panel. An example follows:

`http://www.example.com/magento/admin`

(The Admin Panel URL is a combination of the **Base URL** and **Admin Path** fields you entered when you installed Magento.

- b. Log in to the Admin Panel as an administrator.

- c. Scroll to the bottom of the page; the version should display as 1.14 (Magento EE) or 1.9.0.0 (Magento CE).  
The following figure shows an example.



Congratulations! You successfully installed Magento!

## Setting Up Cronjobs

Several Magento features require at least one cronjob, which schedules activities to occur in the future. A partial list of these activities follows:

- Catalog price rules
- Newsletters
- Generating Google sitemaps
- Customer Alerts/Notifications (product price change, product back in stock)
- Reindexing (*Magento EE 1.13 only*)
- Private sales (*Magento EE only*)
- Automatic updating of currency rates
- *Magento EE 1.14.1 and later, Magento CE 1.9.1 and later* All Magento e-mails (including order confirmation and transactional)



**Note:** Magento depends on proper cronjob configuration for many important system functions, including indexing.  
Failure to set it up properly means Magento won't function as expected.

Crontab files define tasks (cronjobs) that are performed at scheduled dates and times. Each user on a system has its own crontab file, and the cron daemon runs each cronjob as the user who owns the crontab. For Magento, this user is the web server.

## Recommendations for cronjobs

Magento recommends splitting your cronjob on two tasks in your crontab for best performance and completing your cronjobs without issues:

- Indexer Cronjob (`malways` mode) can take longer to execute and complete, skipping other cronjobs (`mdefault` mode) scheduled during the time that the indexer runs.
- Cronjobs in `mdefault` scope may take longer to execute and complete, blocking others scheduled during the time it is still running.

To split your `malways` and `mdefault` cronjob modes, use the following:

```
* * * * * /path/to/magento/cron.sh cron.php -mdefault
* * * * * /path/to/magento/cron.sh cron.php -malways
```

Determine which of your cronjobs may be long-running and move them into separate crontabs. For example, imports, exports, and indexes can run long, blocking other cronjobs scheduled in that crontab.

Magento recommends running cron every minute for EE and every five minutes for CE.

## Create cronjobs

First, determine your web server's user. SSH into the server and enter the following command:

```
ps -o "user group command" -C httpd,apache2
```

In CentOS, the Apache user is typically `apache`. In Ubuntu, it's typically `www-data`. Enter this user in the commands for creating the crontabs and cronjobs.

We recommend splitting the cronjob between two tasks to prevent the indexer blocking other `mdefault` jobs or an `mdefault` job blocking the indexer. These instructions include these commands.

To create a cronjob as the user that runs Apache:

- a. Create or edit a crontab for the Apache user. This command opens a vim text editor with the username of the Apache user. You might need to choose a text editor first.

```
crontab -u apache-user-name -e
```

- b. To split cronjobs for better performance, enter the following:

```
* * * * * /bin/sh /path/to/magento/cron.sh cron.php -mdefault
* * * * * /bin/sh /path/to/magento/cron.sh cron.php -malways
```

For example, for CentOS,

```
* * * * * /bin/sh /var/www/html/magento/cron.sh cron.php -mdefault
* * * * * /bin/sh /var/www/html/magento/cron.sh cron.php -malways
```

- c. If you do not want to split the cronjob, enter the following:

```
* * * * * /bin/sh /path/to/magento/cron.sh
```

For example, for CentOS,

```
* * * * * /bin/sh /var/www/html/magento/cron.sh
```

- d. Save your changes to the crontab and exit the editor. For vim, press :wq.

To check if crons are running, you should see the following log files in the Magento directory about 10 minutes after the scheduled cron runs:

```
./cron.php.log
./cron.sh.log
```

## Setting Magento File System Permissions and Ownership After Installation

To secure your Magento installation after the installation, see [After You Install Magento: Recommended File System Ownership and Privileges](#).

## Troubleshooting

This section discusses some suggestions for issues you might encounter when installing Magento.

| ISSUE                                                                                                                 | SUGGESTED SOLUTION                                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| During the installation, errors like the following display:<br>Path "/var/www/html/magento/app/etc" must be writable. | Set privileges and ownership on the Magento installation directory and subdirectories as discussed in <a href="#">Setting File and Directory Ownership and Privileges</a> . |
| You never advance past the installer's Configuration page.                                                            | Try selecting the <b>Skip Base URL Validation Before the Next Step</b> checkbox. If problems persist, <a href="#">get help</a> .                                            |

22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Upgrade Roadmap for Magento Community Edition (CE) 1.8 or 1.9

Magento recommends you upgrade your installation using the following guidelines in a *development or test environment*, separate from your existing production environment:

1. Install Magento in a different directory:
  - **Recommended.** Set up a new system (that is, another host) on which to install Magento. The system should be identical to, if not better than, your current system. The new system must meet the [Magento system requirements](#).
  - Install Magento in a new, empty root installation directory on the same server.
2. In your current production environment:
  - a. Back up your Magento database.
  - b. Archive the file system.  
This includes the `media` directory and subdirectories; all extensions and customizations; and all custom themes.
3. In the development or test environment:
  - a. Create a new, empty database instance.
  - b. Import the production database tables into the development database instance.
  - c. Copy your production `media` directory, extensions, themes, and other customizations to the development system.
  - d. Copy `local.xml` to `[your Magento install dir]/app/etc` and edit it if necessary to reference the production database instance.
  - e. In a web browser, go to your development system base URL.
  - f. Wait for upgrade scripts to run.
4. *Test the development system thoroughly*, including:
  - Verify all extensions, themes, and customizations work.
  - Place orders using all webstores and all payment methods.

When you're ready to start your upgrade, see [Upgrading to and Verifying Magento Community Edition 1.8 and Enterprise Edition 1.13 - Part 2](#).

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# EE 1.14 Upgrade: Running the EE 1.13.0.0 or EE 1.13.0.1 URL Redirect Script

[Edit this page on GitHub](#) 

This article is part of the Magento Enterprise Edition (EE) upgrade documentation. If you're looking for something else, [click here](#) to go back to the Magento Knowledge Base.

This article applies to the following Magento EE upgrades:

- From EE 1.13.0.1 to EE 1.13.0.2
- From EE 1.13.0.1 to EE 1.13.1.0
- From EE 1.13.0.1 to EE 1.14.0.0

If your upgrade is not in the preceding list, skip this article and continue with [Completing the Magento Upgrade](#).

This article discusses how to run the URL redirect script that creates HTTP 301 (Moved Permanently) redirects for any duplicate product URL keys. This enables **Previous** links in your web store to work, for example.

For more information about URL changes, see the [EE 1.13.0.2 Release Notes](#). (Release Notes for the latest release are [here](#).)

To create redirects for your URLs, perform the following tasks in the order shown:

1. [Verifying that cron is Disabled](#)
2. [Running the Script](#)

## Verifying that cron is Disabled

It's very important to stop cron jobs from running until after the upgrade script completes. To do so, enter one of the following commands as a user with `root` privileges:

- Ubuntu: `service crond stop`
- CentOS: `service cron stop`

## Running the Script

This section discusses how to run the URL redirect script to create permanent redirects for any URLs that changed in EE 1.13.1.0.

**Note:** The time the script takes to run and the amount of memory it uses is directly proportional to the size of your database.

To run the script:

1. Change to your Magento installation directory.
2. As the user who owns the files (typically, either `root` or the web server user), enter the following command:  
`php -f shell/url_migration_from_1_13_0_0_to_1_13_0_2.php`

3. Wait while the script runs.

Messages like the following display to indicate progress:

```
[INFO]: Initialization...
[INFO]: Start url rewrites processing from 1.13.0.0 to 1.13.0.2 ...
[INFO]: Start root category "Default Category" processing ...
[INFO]: Start root category "test" processing ...
[INFO]: Executed in time
```

4. If the script succeeded, continue with [Completing the Magento Upgrade](#).

### Troubleshooting

If errors display, check [your Magento install dir]/shell/migration.log for exceptions. Following are errors that might display when you run the migration script:

- Application is not installed yet, please complete install wizard first
- PHP Fatal error: Uncaught exception 'PDOException' with message 'SQLSTATE[28000] [1045] Access denied for user 'name'@'hostname' (using password: YES)' in /var/www/magento/lib/Zend/Db/Adapter/Pdo/Abstract.php

If one of these displays, make sure you copied [your Magento install dir]/app/etc/local.xml from your development system to the production system. Edit it if necessary to reference the production database instance. After you copy and edit local.xml, run the URL redirect script again.

Continue your upgrade with [Completing the Magento Upgrade](#).

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# EE 1.14 Upgrade: Running the EE 1.12 and Earlier URL Redirect Script

This article is part of the Magento Enterprise Edition (EE) upgrade documentation. If you're looking for something else, [click here](#) to go back to the Magento Knowledge Base.

This article applies to the following Magento EE upgrades:

- From EE 1.12.x or earlier to EE 1.13.0.2
- From EE 1.12.x or earlier to EE 1.13.1.0
- From EE 1.12.x or earlier to EE 1.14.0.0

If your upgrade is not in the preceding list, go back to [Upgrading to and Verifying Magento Community Edition \(CE\) 1.8 and Enterprise Edition \(EE\) 1.13](#).

This article discusses how to run the URL redirect script that creates HTTP 301 (Moved Permanently) redirects for any duplicate product URL keys. This enables **Previous** links in your web store to work, for example.

To create redirects for your URLs, perform the following tasks in the order shown:

1. [Verifying that cron is Disabled](#)
2. [Changing Indexer Settings to Update When Scheduled](#)
3. [Running the URL Redirect Script](#)
4. [Running a Full Reindex from the Command Line](#)
5. [Viewing the List of Redirects](#)
6. [Verifying That URLs Work Properly](#)
7. [Completing Your Upgrade](#)

## Verifying that cron is Disabled

It's very important to stop cron jobs from running until after the upgrade script completes. To do so, enter one of the following commands as a user with **root** privileges:

- Ubuntu: `service crond stop`
- CentOS: `service cron stop`

## Changing Indexer Settings to Update When Scheduled

To change indexer settings:

1. Click **System > Configuration**.
2. In the left navigation bar, in the ADVANCED group, click **Index Management**.
3. In the right pane, expand **Index Options**.

- Click **Update when scheduled** from the list for every indexer.

The following figure shows an example.

The screenshot shows the 'Index Options' section of the Admin Panel's Index Management page. It lists eight index types, each with a dropdown menu set to 'Update when scheduled'. To the right of each dropdown is the word '[GLOBAL]'. At the top right of the form is a blue 'Save Config' button.

| Index Type                     | Setting               | Scope    |
|--------------------------------|-----------------------|----------|
| Catalog Category/Product Index | Update when scheduled | [GLOBAL] |
| Category URL Rewrite           | Update when scheduled | [GLOBAL] |
| Product URL Rewrite            | Update when scheduled | [GLOBAL] |
| Redirect URL Rewrite           | Update when scheduled | [GLOBAL] |
| Category Flat Index            | Update when scheduled | [GLOBAL] |
| Catalog Search Index           | Update when scheduled | [GLOBAL] |
| Product Flat Index             | Update when scheduled | [GLOBAL] |
| Price and Stock Index          | Update when scheduled | [GLOBAL] |

- In the upper right corner, click **Save Config**.



**Note:** Ignore any messages requesting you to reindex or flush the cache; you'll do that later.

- Log out of the Admin Panel.

- If the script ran without errors, continue with [Running the URL Redirect Script](#).

## Running the URL Redirect Script

This section discusses how to run the URL redirect script to create permanent redirects for any URLs that changed in EE 1.13.1.0.



**Note:** The time the script takes to run and the amount of memory it uses is directly proportional to the size of your database. If your database has 100,000 SKUs or more, Magento *strongly recommends* you monitor thread execution as discussed in this section. For large databases, the script can require several hours to complete.

To run the script:

- Change to your Magento installation directory.
- As the user who owns the files (typically, either `root` or the web server user), enter the following command:

```
php -f shell/url_migration_to_1_13.php - thread-count
```

where

`thread_count` is the number of threads to use. Magento recommends setting this to the number of CPU cores in your system minus 1 to a maximum of 15. For example, if your system has 8 CPU cores, enter

```
php -f shell/url_migration_to_1_13.php 7
```



**Note:** Each thread can use up to 512MB of RAM.

- Wait while the script runs.

Messages like the following display to indicate progress:

```
[INFO]: Initialization...
[INFO]: Renaming conflicting entities...
100% [#####
[INFO]: Creating redirects from previous version...
100% [#####
ETA HH:MM:SS
```

- If you have a large database, open another command window and run the following command:

```
watch -n1 "ps ax | grep umt | grep -v grep"
```

This command helps you to determine if each thread behaves correctly. After each process finishes, a new process should start immediately. There should be no delays or "hangs" between processes.

- If the script runs successfully, continue with [Running a Full Reindex from the Command Line](#).

If errors display, check `[your Magento install dir]/shell/migration.log` for exceptions and see the following table.

### Troubleshooting

| SYMPTOM                                               | SUGGESTED ACTION                                                                                                                                                                                    |
|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERROR: Scope of attribute "url_key" is set to Global. | This error can be caused by either a product or category <code>url_key</code> attribute being set for Global scope. To determine which type of <code>url_key</code> attribute is causing the error: |

This may cause DB inconsistency. Aborting.

- Log in as an administrator to the Admin Panel of the production system (in other words, the system you're upgrading *from*).

- Click **Catalog > Attributes > Manage Attributes**.

- Click the **url\_key** row.

- Examine the value of the **Scope** list. A sample follows.

### Edit Product Attribute "URL Key"

| Attribute Properties               |                                                                                                                                                             |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Attribute Code *                   | <input type="text" value="url_key"/><br>▲ For internal use. Must be unique with no spaces.<br>Maximum length of attribute code must be less than 30 symbols |
| Scope                              | <input type="text" value="Global"/><br>▲ Declare attribute value saving scope                                                                               |
| Catalog Input Type for Store Owner | <input type="text" value="Text Field"/>                                                                                                                     |
| Default Value                      | <input type="text"/>                                                                                                                                        |
| Unique Value                       | <input type="text" value="No"/><br>▲ Not shared with other products                                                                                         |

- If the value in the **Scope** list is **Global**:

- Select another value from the list. You can select any value *other than Global*.

- Click **Save Attribute**.

- Log out of the Admin Panel.

- Re-export the database as discussed in [Backing Up the Database](#).

- In the development system, drop and re-create the database as discussed in [Pre-Upgrade Tasks for Your Development Environment](#).

- Import the database you just exported as discussed in [Updating the Magento Database](#).

- Try running the upgrade script again.

- If the value of the **Scope** list is *not Global*, the issue is likely that the category url\_key attribute is set to global. To confirm this:

- Log out of the Admin Panel.

- In the development database instance, use either the MySQL command line or phpmyadmin to get the value of **is\_global** from the **catalog\_eav\_attribute** table. If the value is 1, your catalog url\_key attribute is set to Global.

Following is an example of finding the value using the MySQL command line:

```
select attribute_id, is_global from catalog_eav_attribute /
where attribute_id in (select attribute_id from eav_attribute /
where attribute_code = 'url_key');
```

**Note:** The command must be entered on a single line. It is shown here on multiple lines because of space limitations. When you enter the command, omit the / characters.

- If any returned value is 1, you must contact [Magento Support](#).

Other errors

Try running the script again. If errors persist, contact [Magento Support](#).

## Running a Full Reindex from the Command Line

After successfully running the URL redirect script, enter the following command from the Magento root directory:

```
php -f shell/indexer.php --reindexall
```

This command runs a full reindex and it might take a long time, depending on the size of your database.

After the reindex completes, enter the following command from your Magento installation directory:

```
rm -rf var/cache var/full_page_cache var/locks
```

## Viewing the List of Redirects

As a result of running the URL redirect script, a list of redirects created for duplicate URL keys displays in the Admin Panel. To view these redirects:

1. Log in to the Admin Panel as an administrator.

2. Click **Catalog > URL Redirects**.

3. To optionally sort results by description, click the **Description** column.

All redirects created by the URL redirect script display the following description:

1.12.0.2-1.13.x migration redirect

The following figure shows an example.

| ID  | Store View                                                  | Request Path     | Target Path                                   | Options | Description                        | Action               |
|-----|-------------------------------------------------------------|------------------|-----------------------------------------------|---------|------------------------------------|----------------------|
| 260 | testwebsite<br>test                                         | test-15.html     | test-0-9dfcfef58dfe04af37f137a1d9d3e5.html    | RP      | 1.12.0.2-1.13.x migration redirect | <a href="#">Edit</a> |
| 258 | Main Website<br>Main Website Store<br>test                  | test-1-2-11.html | test-1-2-3.html                               | RP      | 1.12.0.2-1.13.x migration redirect | <a href="#">Edit</a> |
| 255 | Main Website<br>Main Website Store<br>Default Store<br>View | test-1-21.html   | test-7-698d51a19db1a121ca581499d75701668.html | RP      | 1.12.0.2-1.13.x migration redirect | <a href="#">Edit</a> |

## Verifying That URLs Work Properly

Assuming all your post-upgrade issues are fixed, go to your web store and navigate the category tree. Click some products and make sure they display properly. Verify that all **Previous** and **Back** links work properly.

## Completing Your Upgrade

Complete your upgrade:

1. [Setting Up the Magento Cron Job](#)
2. [Clearing Magento var/ Subdirectories](#)
3. Complete the remainder of the tasks on that page in the order shown

22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# Ubuntu: Installing Required Prerequisite Software for Magento CE 1.8 Magento EE 1.13 (or Later)

Edit this page on GitHub 

## Contents

- [Overview](#)
- [Updating System Software](#)
- [Apache](#)
- [nginx](#)
- [PHP](#)
- [MySQL](#)
- [Next Steps](#)

## Overview

This article discusses how to install required prerequisite software for Ubuntu. You must complete these tasks before you install Magento CE 1.8 or later or Magento EE 1.13 or later.

Before you continue, make sure you familiarize yourself with the installation process discussed in [Prerequisites](#).



**Note:** You must install system software on Ubuntu as a user with `root` privileges.

## Updating System Software

It's a good practice to update your repositories and system software, if necessary.

Log in to your Magento server as a user with `root` privileges and enter the commands shown in this section.

Update repositories:

```
apt-get update
```

Optionally upgrade software. This might require a system reboot.

```
apt-get upgrade
```

# Apache

This section discusses how to install Apache. For more details, you can consult a reference like the [Ubuntu site](#).

Magento requires Apache use server rewrites. You must also specify the type of directives that can be used in `.htaccess`, which Magento uses to specify rewrite rules.

Installing and configuring Apache is basically a three-step process: install the software, enable rewrites, and specify `.htaccess` directives.

## Installing Apache

Install Apache 2 if you haven't already done so:

```
apt-get -y install apache2
```

## Enabling Apache Rewrites

Ubuntu 12 (which natively supports Apache 2.2) is different from Ubuntu 14 (which natively supports Apache 2.4).

It's very important you choose a value for `AllowOverride` that is suited to your deployment. You can use `AllowOverride All` in development but it might not be desirable in production.

For more information, see one of the following references:

- [Apache 2.2](#)
- [Apache 2.4](#)

### Enabling Apache Rewrites for Apache 2.2

Use this section to enable Apache rewrites and specify `.htaccess` if you use Apache 2.2, which is supported by the default Ubuntu 12 repository.

1. Open the following file for editing.

```
vim /etc/apache2/sites-available/default
```

2. Locate the following block.

```
<Directory /var/www/>
 Options Indexes FollowSymLinks MultiViews
 AllowOverride None
 Order allow,deny
 allow from all
</Directory>
```

3. Change the value of `AllowOverride` to [value from Apache site].

```
<Directory /var/www/>
 Options Indexes FollowSymLinks MultiViews
 AllowOverride All
 Order allow,deny
 allow from all
</Directory>
```

4. Save the file and exit the text editor.

5. Configure Apache to use the `mod_rewrite` module.

```
cd /etc/apache2/mods-enabled
ln -s ../mods-available/rewrite.load
```

6. Restart Apache.

```
service apache2 restart
```

### Enabling Apache Rewrites for Apache 2.4

Use this section to enable Apache rewrites and specify `.htaccess` if you use Apache 2.4, which is supported by the default Ubuntu 14 repository.

1. Enter the following command:

```
a2enmod rewrite
```

2. Specify the type of directives that can be used in `.htaccess`.

For guidelines, see the [Apache 2.4 documentation](#).

Note that in Apache 2.4, the server's default site configuration file is `/etc/apache2/sites-available/000-default.conf`

For example, you can add the following to the bottom of `000-default.conf`:

```
<Directory "/var/www">
AllowOverride [value from Apache site]
</Directory>
```

**Note:** You must change the value of `AllowOverride` in the directive for the directory to which you expect to install the Magento software. For example, to install in the web server docroot, edit the directive in `<Directory /var/www>`.

3. Restart Apache:

```
service apache2 restart
```

## PHP

Magento CE and EE support the following PHP versions:

- Magento CE 1.6.0.0–1.8.1.0 and Magento EE 1.11.0.0–1.13.1.0 support PHP 5.3 natively. They can be used with PHP 5.4 if you apply the [PHP 5.4 patch](#).
- Magento CE 1.9.0.x and EE 1.14.0.x support PHP 5.4 natively. They are backward-compatible with PHP 5.3.
- Magento CE 1.9.1 and EE 1.14.1 support PHP 5.5 natively. They are backward-compatible with PHP 5.4.

We recommend you use the most recent PHP version supported by your version of Magento. For example, you should use PHP 5.5 with CE 1.9.1 or EE 1.14.1.

For more information about PHP, see:

- [PHP 5.4 changelog](#)
- [PHP 5.5 changelog](#)

Check with a system administrator or reference for your version of Ubuntu to see what PHP versions are available.

If you're installing Magento CE 1.9.1 or Magento EE 1.14.1, you can use PHP 5.5; otherwise, we recommend PHP 5.4. For certain versions of CE and EE, a [patch](#) is required to use PHP 5.4.

Enter the following command to see what version of PHP is currently running:

```
php -v
```

See one of the following sections for more information:

- [Upgrading PHP](#)
- [Installing PHP](#)
- [Installing Required PHP Extensions](#)
- [Increasing the PHP memory Limit](#)

## Upgrading PHP

**!** **Important:** The following sections discuss *suggested* PHP upgrade paths. Because you're choosing a non-Ubuntu repository to upgrade PHP, make your choice carefully. Not all repositories work equally well. We don't recommend any particular repository. Consult a system administrator or Ubuntu reference for more information.

To upgrade your version of PHP, see one of the following:

- Upgrade to PHP 5.5: [PHP site](#) and [dev-metal](#).
- Upgrade to PHP 5.4: [phpave](#).

## Installing PHP

See one of the following sections:

- [Installing PHP 5.5 on Ubuntu 12](#)
- [Installing PHP 5.4 on Ubuntu 12](#)
- [Installing PHP 5.5 on Ubuntu 14](#)

## Installing PHP 5.5 on Ubuntu 12

Use the following commands from [dev-metal](#):

```
add-apt-repository ppa:ondrej/php5
apt-get update
sudo apt-get install php5
```

## Installing PHP 5.4 on Ubuntu 12

Use the following commands from [askubuntu](#) :

```
add-apt-repository ppa:ondrej/php5-oldstable
apt-get update
apt-get upgrade
apt-get install php5
```

## Installing PHP 5.5 on Ubuntu 14

Enter the following command:

```
apt-get -y install php5
```

## Installing Required PHP Extensions

Enter the following command to install PHP extensions required by Magento:

```
apt-get -y install php5 php5-mhash php5-mcrypt php5-curl php5-cli php5-mysql php5-gd libapache2-mod-php5
```

## Increasing the PHP memory Limit

Increase `memory_limit` in `php.ini` to at least 512MB:

1. Open `/etc/php5/apache2/php.ini` in a text editor.

2. Change `memory_limit` to:

```
memory_limit = 512M
```

3. Save your changes and exit the text editor.

## MySQL

Magento CE and EE support the following MySQL versions:

- Magento CE 1.9.1 and Magento EE 1.14.1 support MySQL versions 5.0.2 through 5.6.x.  
To install MySQL 5.6, see [Installing MySQL 5.6](#).
- Magento CE versions 1.8.0.0–1.9.0.x support MySQL versions 4.1.20–5.5.x.  
To install MySQL version 5.5, see the next section.

## Installing MySQL 5.5

Install the MySQL database:

```
apt-get -y install mysql-client mysql-server
```

## Installing MySQL 5.6

Only Magento CE 1.9.1 and EE 1.14.1 support MySQL 5.6. To install MySQL 5.6 on Ubuntu 14, see [Installing MySQL 5.6 on Ubuntu 14](#). To install MySQL 5.6 on Ubuntu 12, see the next section.

## Installing MySQL 5.6 on Ubuntu 12

To install MySQL 5.6 on Ubuntu 12, enter the following commands in the order shown:

```
apt-add-repository ppa:ondrej/mysql-5.6
apt-get -y update
apt-get -y install mysql-server
```

Test the installation by entering the following command:

```
mysql -u root -p
```

Messages similar to the following display:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 5.6.21-1+deb.sury.org~precise+1 (Ubuntu)
```

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

## Installing MySQL 5.6 on Ubuntu 14

To install MySQL 5.6 on Ubuntu 14, enter the following command:

```
apt-get -y install mysql-server-5.6 mysql-client-5.6
```

Test the installation by entering the following command:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.6.19-0ubuntu0.14.04.1 (Ubuntu)
```

```
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

mysql>

## Next Steps

After installing the required prerequisite software, continue your installation with [Preparing Your Server for Magento CE or EE](#).

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

## Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

# CentOS: Installing Required Prerequisite Software for Magento CE 1.8 Magento EE 1.13 (or Later)

[Edit this page on GitHub](#)



## Contents

- [Overview](#)
- [Updating System Software](#)
- [Apache](#)
- [nginx](#)
- [PHP](#)
- [MySQL](#)
- [Next Steps](#)

## Overview

This article discusses how to install required prerequisite software for CentOS. You must complete these tasks before you install Magento CE 1.8 or later or Magento EE 1.13 or later.

Before you continue, make sure you familiarize yourself with the installation process discussed in [Prerequisites](#).



**Note:** You must install system software on CentOS as a user with `root` privileges.

## Updating System Software

It's a good practice to update your repositories and optionally update system software.

Update repositories:

```
yum -y update
```

Optionally upgrade software. This might cause a system reboot.

```
yum -y upgrade
```

# Apache

Magento requires Apache use server rewrites. You must also specify the type of directives that can be used in `.htaccess`, which Magento uses to specify rewrite rules.

Installing and configuring Apache is basically a three-step process: install the software, enable rewrites, and specify `.htaccess` directives.

## Installing Apache

Install Apache 2 if you haven't already done so.

```
yum -y install httpd
```

## Enabling Apache Rewrites

1. Open `httpd.conf` for editing.

```
vim /etc/httpd/conf/httpd.conf
```

2. Locate the block that starts with:

```
<Directory /var/www/html>
```

3. In that block, change the value of `AllowOverride` to `All`.

4. Save your changes to `httpd.conf` and exit the text editor.

5. Restart Apache.

```
service httpd restart
```

# PHP

Magento CE and EE support the following PHP versions:

- Magento CE 1.6.0.0–1.8.1.0 and Magento EE 1.11.0.0–1.13.1.0 support PHP 5.3 natively. They can be used with PHP 5.4 if you apply the [PHP 5.4 patch](#).
- Magento CE 1.9.0.x and EE 1.14.0.x support PHP 5.4 natively. They are backward-compatible with PHP 5.3
- Magento CE 1.9.1 and EE 1.14.1 support PHP 5.5 natively. They are backward-compatible with PHP 5.4

We recommend you use the most recent PHP version supported by your version of Magento. For example, you should use PHP 5.5 with CE 1.9.1 or EE 1.14.1.

For more information about PHP, see:

- [PHP 5.4 changelog](#)
- [PHP 5.5 changelog](#)

Check with a system administrator or reference for your version of CentOS to see what PHP versions are available.

If you're installing Magento CE 1.9.1 or Magento EE 1.14.1, you can use PHP 5.5; otherwise, we recommend PHP 5.4. For certain versions of CE and EE, a [patch](#) is required to use PHP 5.4.

Enter the following command to see what version of PHP is currently running:

```
php -v
```

See one of the following sections for more information:

- [Upgrading PHP](#)
- [Installing Required PHP Extensions](#)
- [Increasing the PHP memory Limit](#)

## Upgrading PHP

CentOS 6.x repositories have PHP 5.3. This section assumes you use either PHP 5.4 or 5.5. Make sure you understand which version of Magento CE or EE supports the PHP version to which you upgrade.

Before you start, verify you have PHP 5.3 installed:

```
php -v
```

If you already have the desired PHP version installed, you don't have to do anything.

If PHP is not installed, install PHP 5.3 using the following command:

```
yum -y install php php-xml
```

Continue with one of the following sections.



**Important:** The following sections discuss *suggested* PHP upgrade paths. Because you're choosing a non-CentOS repository to upgrade PHP, make your choice carefully. Not all repositories work equally well. We don't recommend any particular repository. Consult a system administrator or CentOS reference for more information.

## Upgrading to PHP 5.5

There is more than one way to upgrade CentOS 6.5 to PHP 5.5; the following is a suggestion only. Consult a reference for additional options.

Enter the following commands in the order shown.

```
cd /tmp
rpm -Uvh https://mirror.webtatic.com/yum/el6/latest.rpm
yum -y remove php-common-5.3.3-40.el6_6.x86_64
yum -y install php55w php55w-ocpache
```

Skip the next section and continue with [Installing Required PHP Extensions](#).

## Upgrading to PHP 5.4

Enter the following commands to upgrade to PHP 5.4:

```
cd /tmp
rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
rpm -Uvh http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
yum --enablerepo=remi install httpd php php-common
```

The following resources are also available:

- [kb.parallels](#)
- [stackoverflow](#)
- [remi repository](#)

## Installing Required PHP Extensions

The names of the PHP extensions you must install depend on the repository from which you installed PHP. Use one of the following commands:

PHP 5.5:

```
yum -y install php55w-xml php55w-mcrypt php55w-gd php55w-devel php55w-mysql php55w-mbstring
```

PHP 5.4:

```
yum -y install --enablerepo=remi php-mcrypt gd gd-devel php-gd php-mysql php54w-mbstring
```

## Increasing the PHP Memory Limit

Increase `memory_limit` in `php.ini` to at least 512MB.

1. Open `/etc/php.ini` in a text editor.

2. Change `memory_limit` to:

```
memory_limit = 512M
```

3. Save your changes and exit the text editor.

## MySQL

This section discusses how to install and configure MySQL 5.6. CentOS 6.x repositories have MySQL 5.1; to install a different version of MySQL, see the [MySQL documentation](#).



**Note:** Use the tasks that follow only on a *new MySQL database*. Some of the tasks require you to delete users and should not be performed on a database that has already been set up.

The following procedure is based on [Install MySQL Server 5.6 in CentOS 6.x and Red Hat 6.x Linux](#).

1. Install the MySQL database:

```
cd /tmp
wget http://repo.mysql.com/mysql-community-release-el6-5.noarch.rpm
rpm -ivh mysql-community-release-el6-5.noarch.rpm
yum -y install mysql-server
```

2. Start MySQL.

```
service mysqld start
```

3. Set a password for the `root` user and set other security-related options. Enter the following command and follow the prompts on your screen to complete the configuration.

```
mysql_secure_installation
```

## Next Steps

After installing the required prerequisite software, continue your installation with [Preparing Your Server for Magento CE or EE](#).

 22 days ago

- [Become a Contributor](#)
- [Glossary](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [License/Trademark FAQ](#)
- [Release Notes](#)
- [Third-Party Licenses](#)

© 2020 Magento. All rights reserved.

### Magento 1.x Software Support Notice

For Magento Commerce 1, Magento is providing software support through June 2020. Depending on your Magento Commerce 1 version, software support may include both quality fixes and security patches. Please review our [Magento Software Lifecycle Policy](#) to see how your version of Magento Commerce 1 is supported.

For Magento Open Source 1.5 to 1.9, Magento is providing software security patches through June 2020 to ensure those sites remain secure and compliant. Visit our [information page](#) for more details about our software maintenance policy and other considerations for your business.

## nginx configuration

### Install nginx

We support nginx version 1.7.x. Installing the nginx software is beyond the scope of this guide. You can refer to a resource like the following:

- [nginx wiki](#)
- [How To Install Nginx on Ubuntu 14.04 LTS \(digitalocean\)](#)
- [How To Install Nginx on CentOS 6 \(digitalocean\)](#)

### nginx security setting

[Byte.nl](#) recently reported that some misconfigured Magento sites using the nginx web server software are vulnerable to attacks. The misconfiguration allows outside access to Magento cache files. The cache files have predictable names and can contain sensitive information, including Magento database passwords. This information can be used to obtain access to an installation and customer information.

To avoid this issue, you can use [this nginx configuration](#) provided by Willem de Groot.

We also recommend you review the [Magento Security Best Practices](#).

Additionally, you can also check your site for other security vulnerabilities at <http://magereport.com>. This is a Magento community project that is not affiliated with Magento.