

# TIPS AND TRICKS

OpenCart 2.0.x



## #2

Special thanks to Daniel Kerr, his team and  
the OpenCart community.

Copyright © 2015 by iSense Ltd.

All rights reserved. No part of this publication text may be  
uploaded or  
posted online without the prior written permission of iSense Ltd.

For permission requests, get in touch with us on  
[sales@isenselabs.com](mailto:sales@isenselabs.com).

ISBN: 978-0-9966004-1-5

# Authors



GEORGE



MIHAIL



ALEXANDER



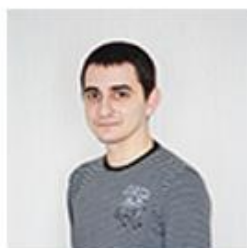
SIMEON



NIKOLA



IVAILO



STOYAN



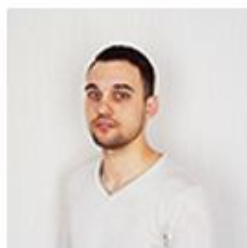
TEYA



JULIAN



JONIDA



EVGENI



ANDREY



VESELIN

# Index

## Introductory

[How to install themes in OpenCart 2.0.x](#)

[How to add a new currency and new language in OpenCart 2.0.x](#)

[How to set up manufacturers in OpenCart 2.0.x](#)

[How to manage taxes in OpenCart 2.0.x](#)

[How to use Google Sitemap in OpenCart 2.0.x](#)

[How to set up a SSL certificate in OpenCart 2.0x](#)

[How to add store credit to customer accounts in OpenCart 2.0.x](#)

[How to find out which OpenCart version am I using](#)

## Tweaks

[Cookie consent Notification in OpenCart 2.0.x](#)

[Customizing your store maintenance mode tutorial](#)

[How to edit the OpenCart footer links](#)

[How to change the “Add to Cart” button name in OpenCart 2.0.x](#)

[How to solve the duplicate content issue in OpenCart 2.0.x](#)

[OCMod stopped working?](#)

## Security

[Block unwanted bots from accessing your site](#)

[How to ban visitors by IP address](#)

## E-commerce Advice

[Seven Ideas for Personalized Customer Experience in eCommerce](#)

[Five beautiful examples of coming soon pages to make customers eager to visit your ready website](#)

[The art of mastering product returns](#)

[The essential guide to good bots, content harvesters and comment spammers](#)

## Bonus

[The Latest OpenCart and Ecommerce News Delivered to your Email Inbox](#)

# How to install themes in OpenCart 2.0?



The release of OpenCart 2 brought a lot of changes to the platform. Probably the most notable among them is the new modern design of the pages. However, for an e-commerce website that is visited by many users everyday, eye-catching design is a must. The theme will not affect the functionality of the website, but an classy one can make a good impression on the customers and make them come back for more shopping. A quick and budget way to achieve this is by applying themes to your OpenCart platform. The Internet is full of great-looking options available, both paid and free.

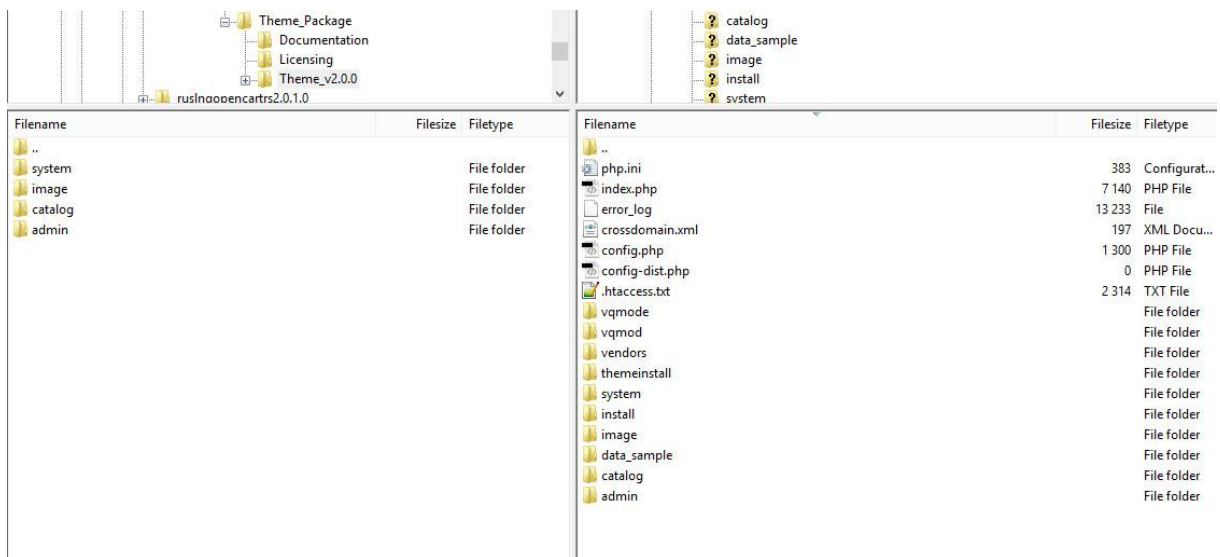
The installation of a theme is a relatively straightforward process and the aim of this tutorial is to guide you through the steps of the process.

1.

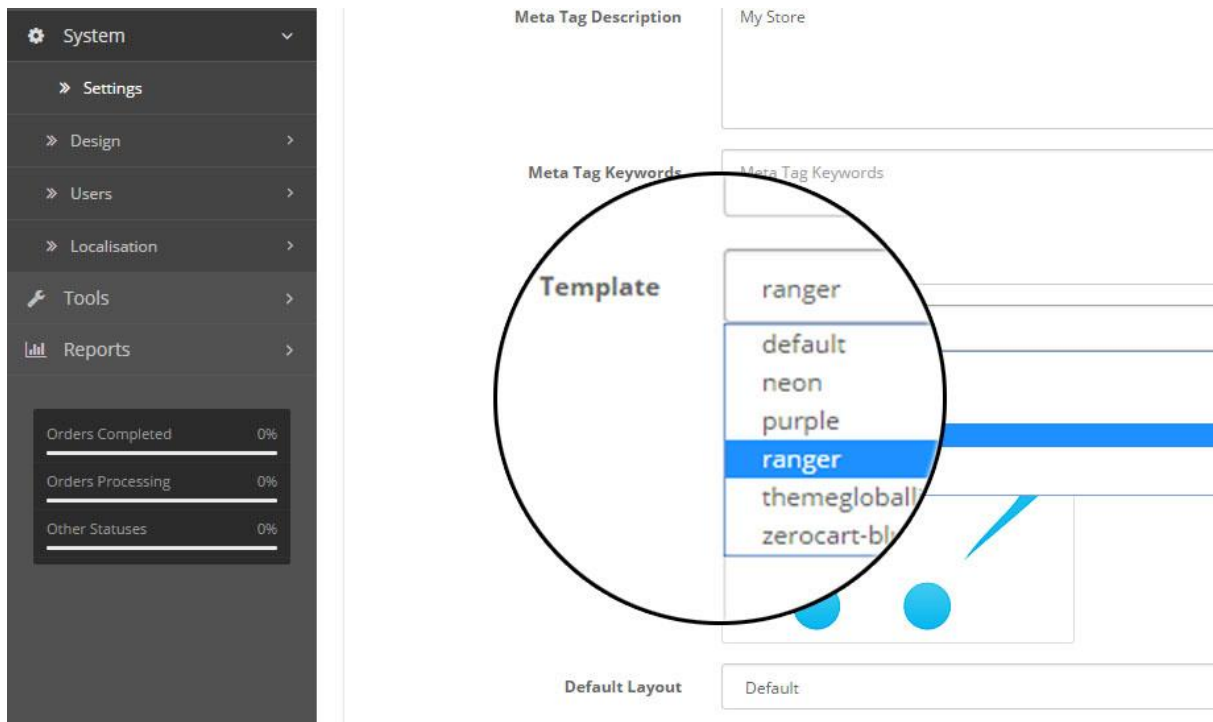
The first step is to find and download a suitable theme for your website. For the purpose of this tutorial we are going to

use the Ranger theme created by KulerThemes and available on <https://isenselabs.com/products/index/opencart/themes>

2. After downloading the archived file, unzip it and launch a FTP client. This can be a standalone program such as FileZilla, SmartFtp, etc. or a built-in client like the one in Adobe Dreamweaver.
3. After connecting to the website, transfer the files from the theme folder into the folder containing the OpenCart installation.



4. Now it is time to login into the admin page of our OpenCart website .Navigate to System/Settings and click on the Edit button next to the name of the store on which to apply the theme.
5. In the settings page, choose the tab Store and look for the label “Template”. From the dropdown menu next to it, the available themes can be seen and chosen from. In my case this is the Ranger theme.



At this point, we have completed installation and have applied the theme to our website. Some themes, however, require some further settings to be made in order to work properly on the website. Information about them is usually stored in the documentation that is contained in the downloaded theme archive.

# How to add a new currency and new language in OpenCart 2.x



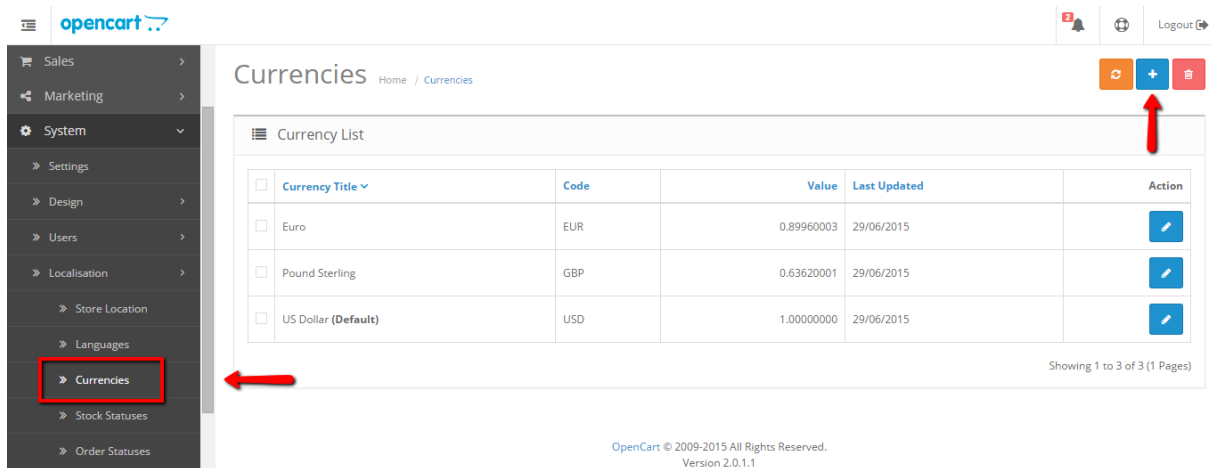
There are a couple of basic things which every OpenCart store owner needs to know when setting up an online store. Among these are adding a currency and a language different than the default OpenCart ones (USD/EUR/GBP and English respectively).

In this blog post we will show you a step-by-step tutorial how you can do these two set ups only from your OpenCart admin panel.

## Adding a New Currency

1. Login to your admin panel and navigate to System >> Localisation >> Currencies and click on the Add new currency button.

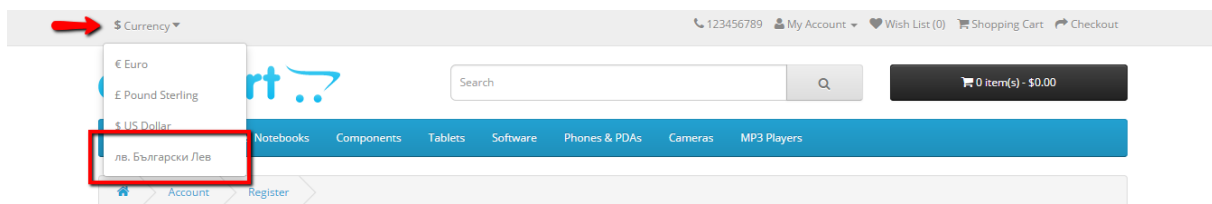




2. After you click the button you will see the form for new a currency. In our example we will be adding Bulgarian lev and the form should be completed in the following manner:



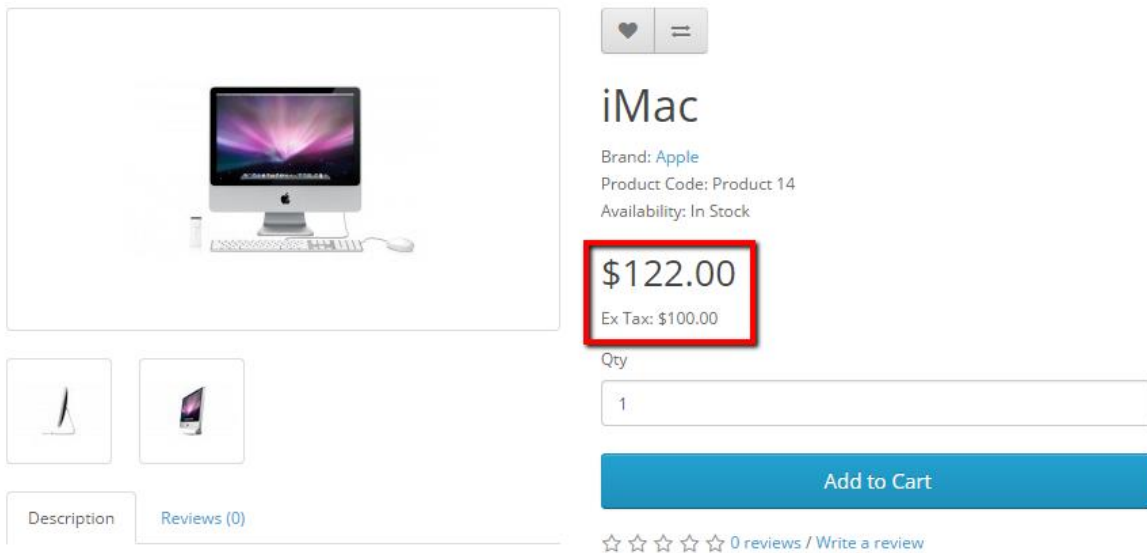
- The **Currency Title** is a required field and its input will be shown in your store front under currencies



- In the **Code field** you will need to add the ISO code of the currency, which you can find here:

<http://www.xe.com/iso4217.php>

- The **Symbol Left** and **Symbol Right** fields are not required, which means that you can leave them empty. However, we would suggest you to add a symbol if your currency has one, as it gives a clearer idea to the customer for the price of a given product. These are two examples of a currency with symbol left:



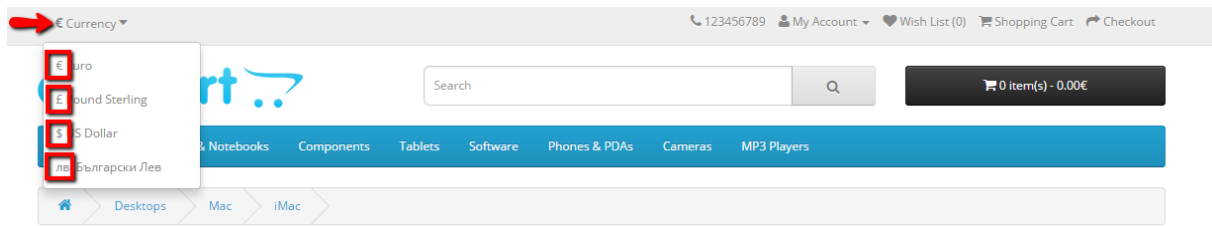
This screenshot shows a product page for an iMac. The price is displayed as \$122.00, with the dollar sign on the left. Below the price, the text 'Ex Tax: \$100.00' is visible. The page includes a product image, a quantity input field set to 1, and an 'Add to Cart' button. The product name 'iMac' and brand 'Apple' are also shown.

and a currency with symbol right:



This screenshot shows a product page for an iMac. The price is displayed as 109.70€, with the Euro symbol on the right. Below the price, the text 'Ex Tax: 89.92€' is visible. The page includes a product image, a quantity input field set to 1, and an 'Add to Cart' button. The product name 'iMac' and brand 'Apple' are also shown.

The left and right symbols are also showing under currencies in the store front:



- The **Decimal Places** field sets how many digits to be displayed after the decimal point. If left blank, the prices will be rounded to a whole number.
- The **Value** field sets the value of the new currency in comparison to the default currency. If you would like your new currency to be the default one, you should type in 1. If you type in any other number or leave the input blank, the value will be calculated automatically. If you, however, wish to set up a value different than the official one, after you have saved the currency, click on the Edit button and type the desired value in the field and click save. In this way, the added by you value will be used when calculating the price.
- After setting the **Status** to 'Enabled' your new currency will be enabled in your store

3. After you have filled in all of the settings, make sure that the ISO code is correct and Save the new currency.

If you have followed the steps carefully, you will get a success message saying that Currencies have been modified. Now is the time to go to your store-front and check if everything is alright with the recently added currency.

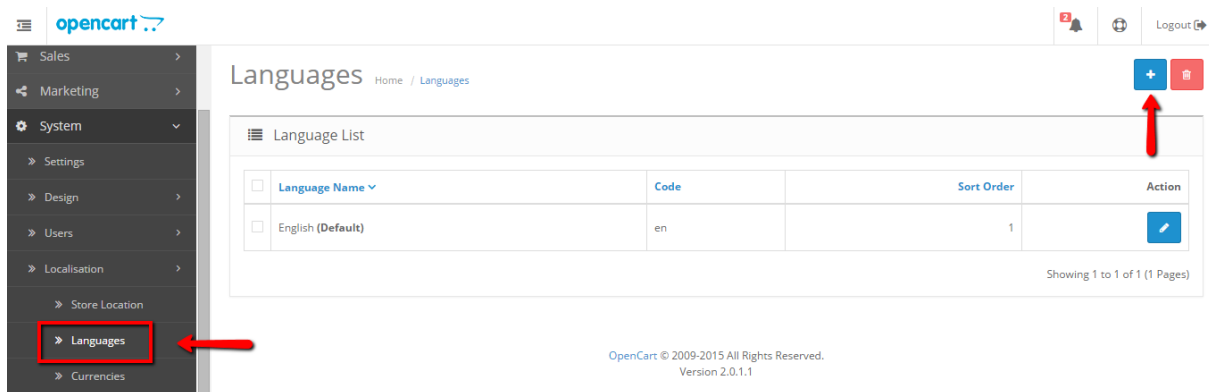
### Auto-update currencies

In general, OpenCart automatically updates the values of the currencies in your store. However, you can disable this setting from System >> Settings >> Store >> Local >> Auto Update Currency >> No.

# Add New Language

In the second part of the tutorial we will demonstrate you how you can add a new language.

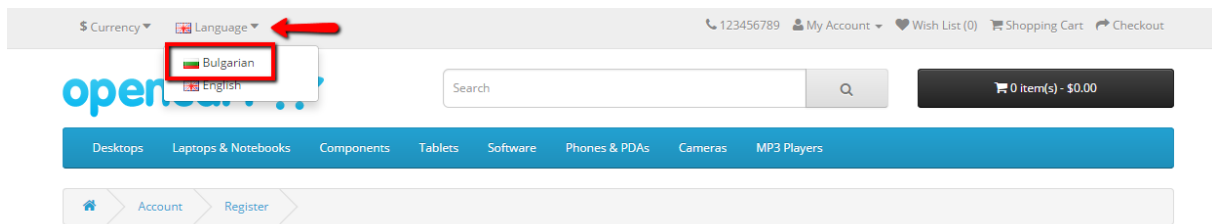
1. Login to your admin panel and navigate to System >> Localisation >> Languages and click on the Add new currency button



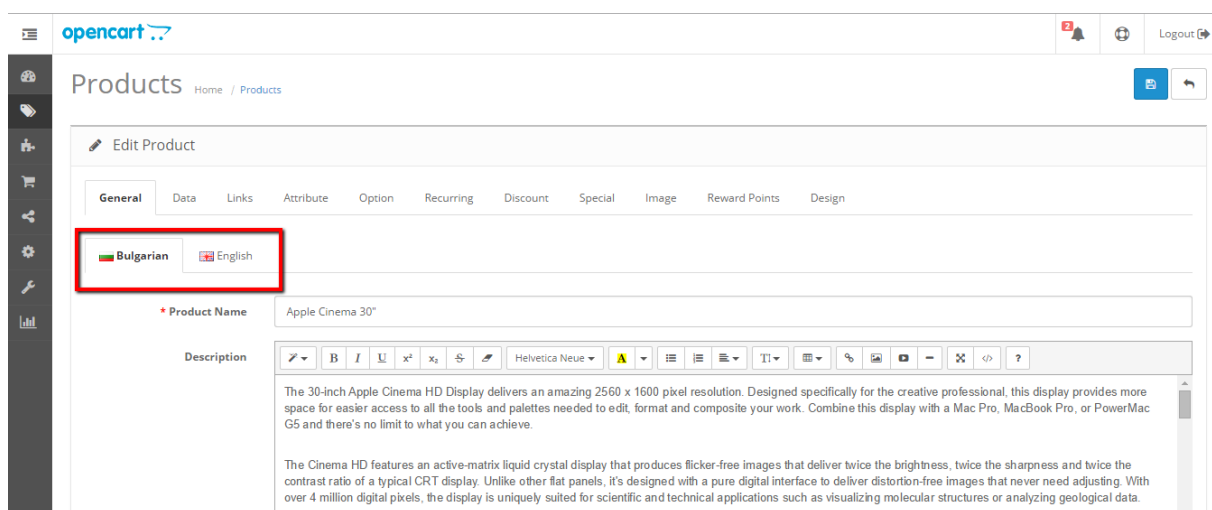
2. An Add-New-Language form will show up. In our example we will be adding Bulgarian language. The form should be completed as following:



- Similarly to the Currency, the **Language name** input field is required and it will be displayed in the front store under languages:



- In the **Language Code** field you have to add the code of the new language which you can find here: [https://www.loc.gov/standards/iso639-2/php/code\\_list.php](https://www.loc.gov/standards/iso639-2/php/code_list.php)
- You can find your **Language Locale** here: <https://developer.chrome.com/webstore/i18n#localeTable>
- The **Image** field sets the image name of the flag of the new language. The flag is displayed in the header of the store-front as well as in most of the text input fields in your admin panel.



The language images come with your OpenCart installation and are in the following folders for the admin and store front respectively: **admin/view/image/flags** and **image/flags**. If you would like to a new image that will be associated with your language, you can upload it in the mentioned folders and specify its name in the image field.

- The **Directory** field specifies the name of the directory in the **admin/language** and **catalog/language** folders, where the language files of your OpenCart website are stored. In the general case, the directory name is the name of the language written with lowercase Latin characters.
- When the **Status** is set to 'Enabled', the new language is enabled in your store.
- The **Sort Order** sets the ordering of the languages.

3. After you have filled in all of the settings, make sure that the language code, image name and directory are correct and Save the new language.

If you have followed the steps carefully, you will get a success message saying that Languages have been modified.

**Note:** Keep in mind that adding a new language through your admin doesn't mean that your store will automatically get translated. In order to learn how you can do that, check our blogpost on "How to add new languages to your OpenCart extensions?"

We hope that you will find this tutorial helpful and easy to follow. Please, address your feedback and questions in the comments section below.

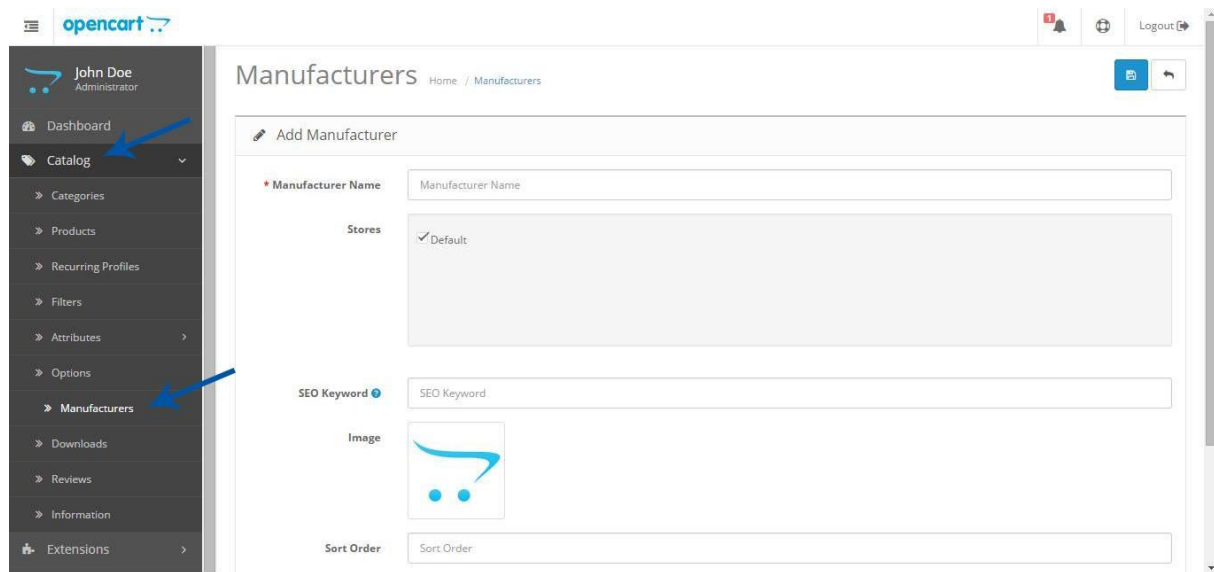
# How to set up manufacturers in OpenCart 2.0.x



In the modern world, a well-known and proven brand means prestige, quality and trustworthiness. A brand is usually the thing that makes up for the price of a given product. Being able to advertise the brands you are selling is a very important part of any marketing strategy. In this article, we will show you how to set up and modify the manufacturers for your OpenCart store.

## How to insert a new manufacturer

- Go to **Catalog** → **Manufacturers**.
- Click on **Add new**.

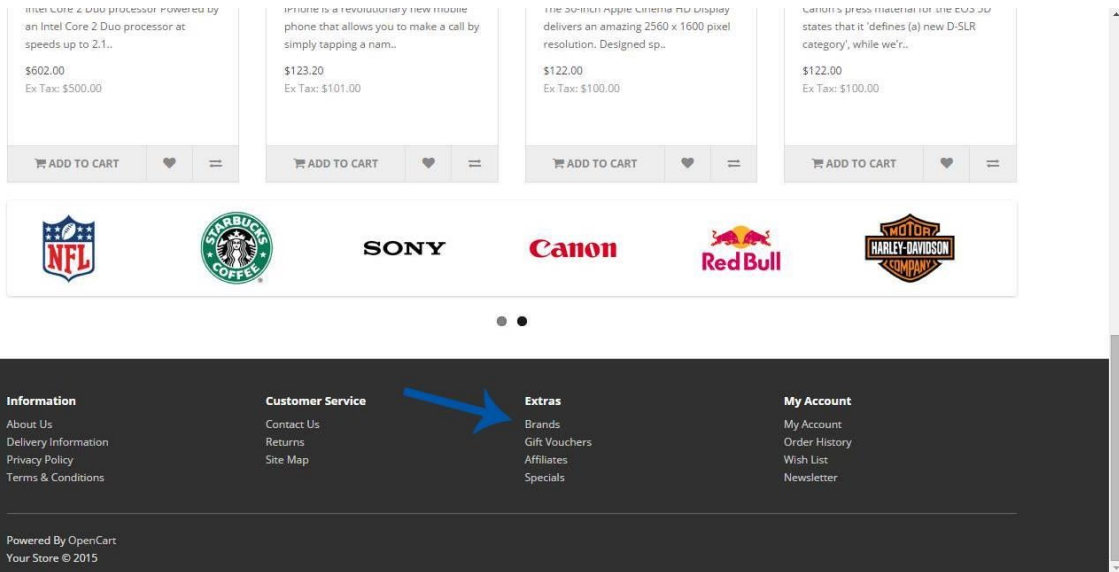


In the form you have to fill in all the information needed for OpenCart to apply the manufacturer. You must specify a name and you have some optional fields as well:

- to select the stores you want it to appear in
- to add some SEO keywords
- to add an image
- to modify the sort order of the manufacturer.

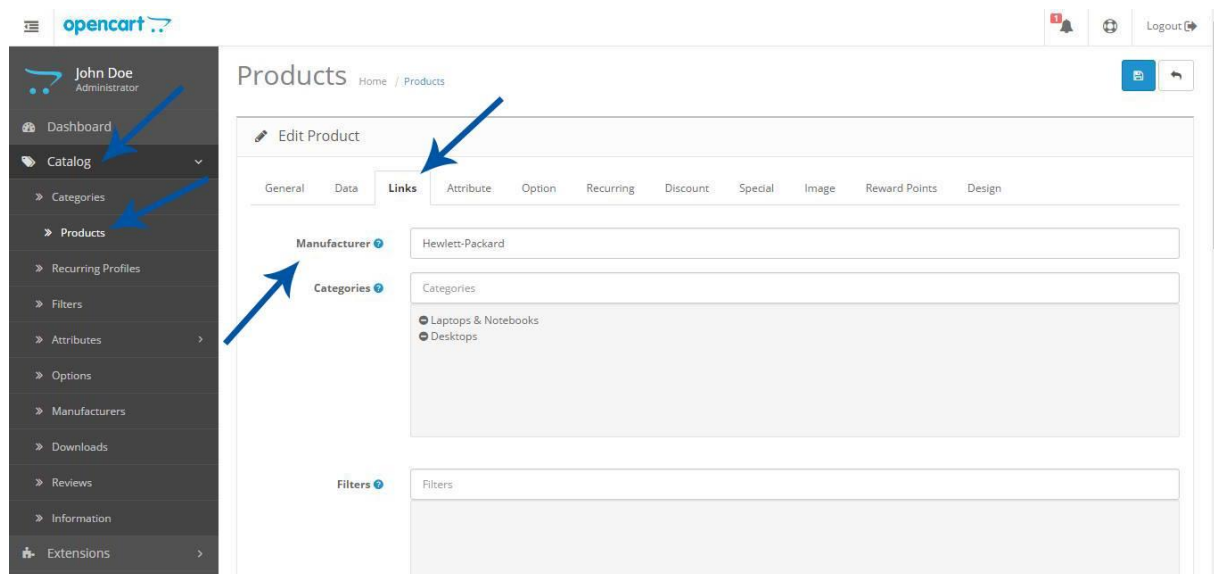
When you click **Save**, you are all set! On the store's front end, a customer can view all your manufacturers when clicking **Brands** on OpenCart's custom footer. An alphabetical list of your brands will appear.





## Adding the manufacturer to your products

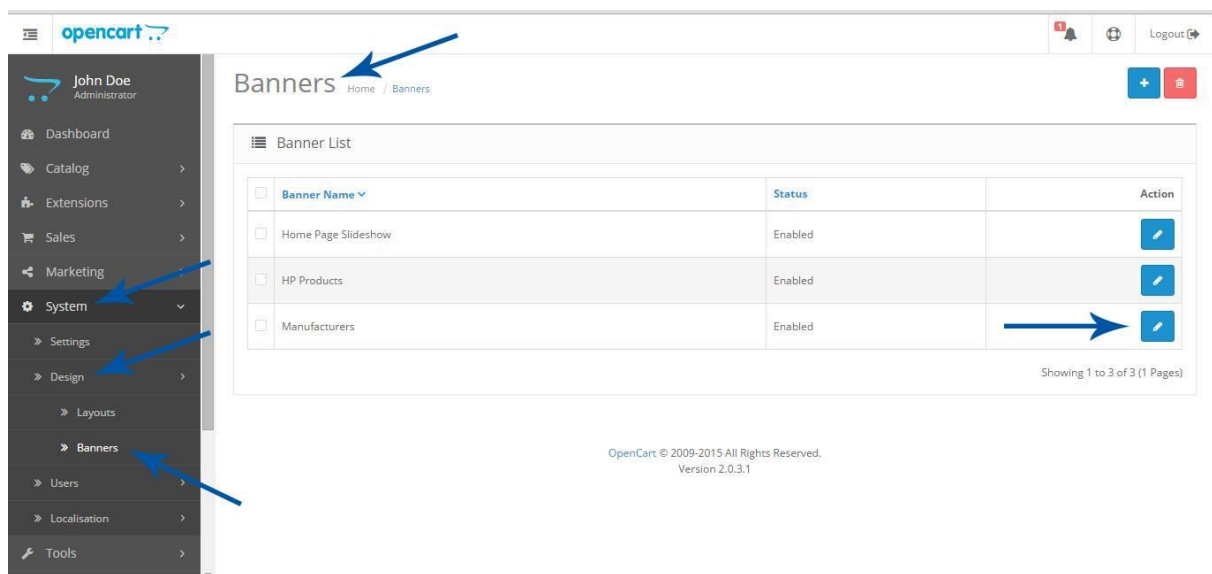
- Go to Catalog → Products.
- Choose a product and click **Edit**.
- Go to **Links**.



The first field you see is labeled **Manufacturer**. This is where you must apply your new manufacturer to the product. When you are done, hit **Save**.

Adding the manufacturer to the 'Manufacturers' banner

- Go to **Brands** on your OpenCart's frontend.
- Click on the brand you added.
- Copy the URL of the page.
- Go to **System** → **Design** → **Banners**
- Click **Edit** on the **Manufacturers** field



- You will see all the currently enlisted manufacturers. Scroll down to the bottom and click on **Add Banner**.
- Type the title of the manufacturer and paste the URL you copied in the **Link** field. After that, upload the logo for you manufacturer.
- Hit **Save**.

You are all set! Go to your store's front end to test the banner's performance.

## Conclusion

Whether it is a custom brand you want to advertise, or a trademark you are working with, which you feel will boost your store's conversion rates, showing it to the public is never a bad thing.

If you want to read up on the *product<>manufacturer* relationship in OpenCart, check out these awesome articles on the subject:

[How to add the manufacturer logo to your product page in OpenCart 2.x](#)

[How to add a tab with products from the same manufacturer in your product page in OpenCart 2.x](#)

If you have any questions, drop us a line below.

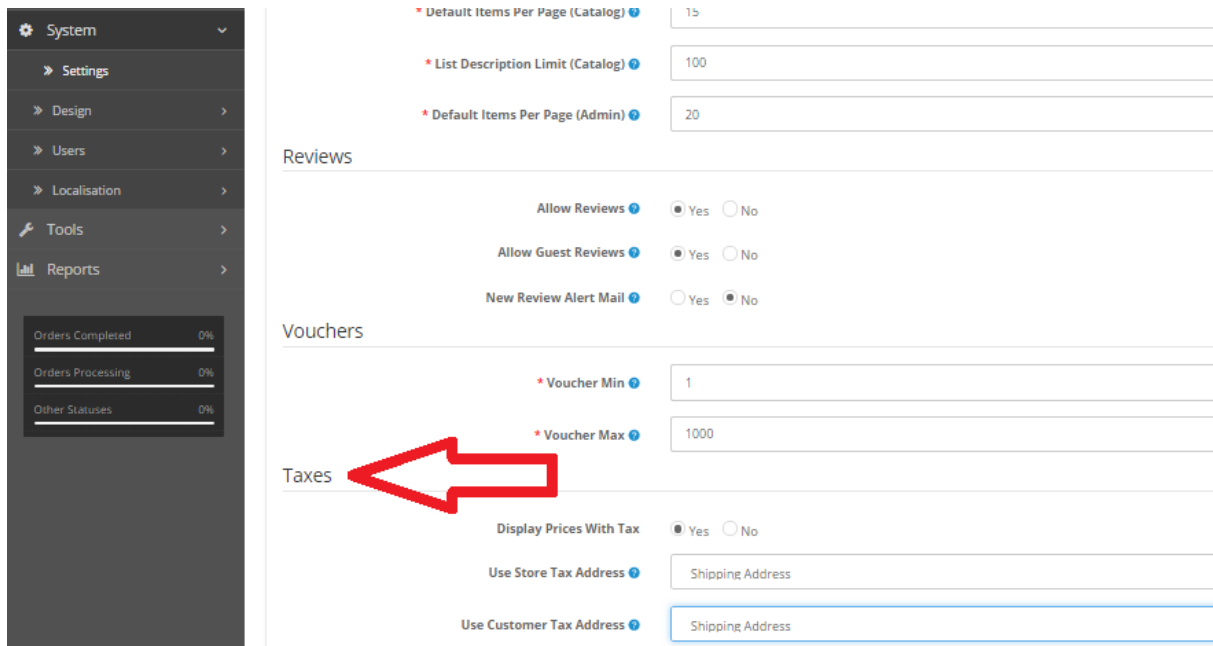
# How to manage taxes in OpenCart 2.0.x

## TAXES

Taxes are crucial component of price formation in many countries and legislations. OpenCart 2.0.x has a feature-rich taxes functionality and in this post I will walk you through the configuration of your taxes.

### Where to find taxing options in OpenCart settings menu?

To set the way taxes are applied to prices go to System->Settings->Options tab and navigate to the fourth section with the heading Taxes.



To modify the taxing settings go to **System->Settings->Localisation**. There you can see the links to Taxes and Geozones. The submenus in Taxes modify the taxing rules, while in Geozones you can determine to which countries and areas to apply them.

## Modifying the settings

When you open **System->Settings->Options** tab, there are three options with which to control the tax applying in your OpenCart website. The group of two radio buttons allow you to switch between showing the product prices with or without taxes. The next two dropdown menus affect the taxing formation when the customer is logged in or logged out. The 'Use Store Tax Address' one uses the store address to calculate taxes if no one is logged in. You can choose to use the store address for the customer's shipping or payment address. The 'Use Customer Tax Address' one allows you to use the customer's default address when they login to calculate taxes. Again, you can choose to use the default address for the customer's shipping or payment address.

In **System->Settings->Localisation->Taxes->Tax Classes**, you can set what type of taxes are applied to each class of products. The menu **System->Settings->Localisation->Taxes->Tax Rates**, gives you the opportunity to set the tax calculation rules-whether it will be a percentage of the whole price or a fixed amount and what the rates are going to be.

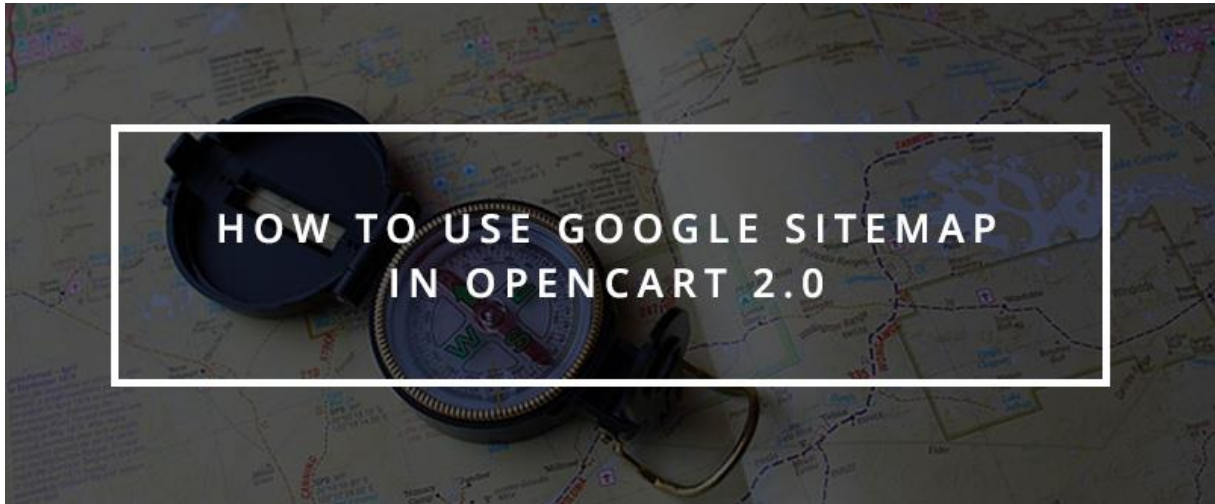
The screenshot shows the 'Tax Classes' management page in OpenCart. On the left is a dark sidebar menu with 'System' expanded to show 'Settings', 'Design', 'Users', 'Localisation', 'Store Location', 'Languages', 'Currencies', 'Stock Statuses', 'Order Statuses', 'Returns', 'Countries', 'Zones', 'Geo Zones', 'Taxes', and 'Tax Rates'. The main content area is titled 'Tax Classes' and includes a breadcrumb 'Home / Tax Classes'. Below this is an 'Edit Tax Class' form. The form has two main sections: 'Tax Class Title' with the value 'Taxable Goods' and 'Description' with the value 'Taxed goods'. Below these is a table with columns for 'Tax Rate', 'Based On', and 'Priority'. The table contains two entries: 'VAT (20%)' based on 'Shipping Address' with priority '1', and 'Eco Tax (-2.00)' based on 'Shipping Address' with priority '2'. Each entry has a red minus icon to its right, and there is a blue plus icon at the bottom right of the table. At the bottom of the page, there is a small copyright notice: 'OpenCart © 2009-2015 All Rights Reserved. Version 2.0.3.1'.

The last option which concerns taxing can be found in **System->Settings->Localisation->Geo Zones**. There you can define geographical zones and then set them to the different taxes back in the Tax Rates menu.

## To sum up

As you can see, managing taxes has been made an easy to understand and deal with task in the new OpenCart version. The possibilities given cover every aspect of taxing and price formation. If there are any questions left unanswered towards tax managing, let us know in the comments section below.

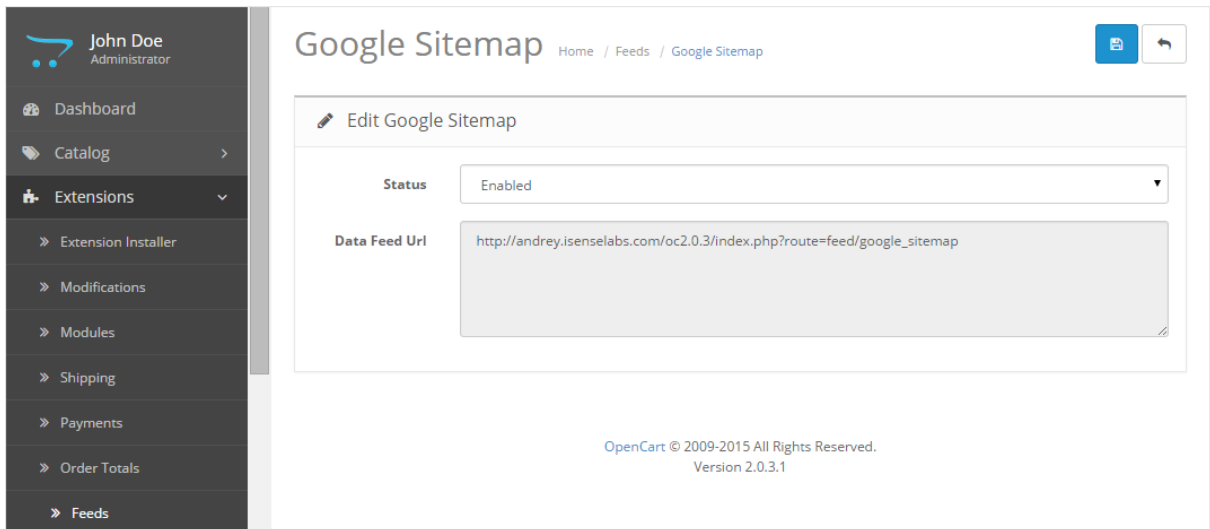
# How to use Google Sitemap in OpenCart 2.0.x



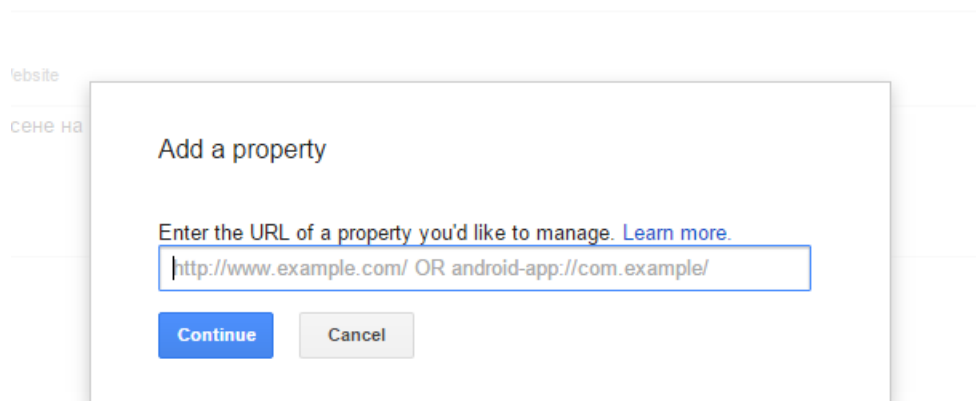
Creating a sitemap of your OpenCart store allows Google and other search engines to crawl more intelligently through the pages of your website. In case your site is linked properly, a map would increase the discoverability of the content of your store. This is why OpenCart has a special feature that makes the whole process of creating a Google sitemap a lot easier and this step-by-step tutorial will guide you through it.

**Here is how to do it:**

1. Login to the Admin panel of your OpenCart store and open the ***Extensions -> Feeds menu***. Install Google Sitemap and enter the Edit Google Sitemap menu. Change the status of the sitemap to enabled and copy the Data Feed URL.



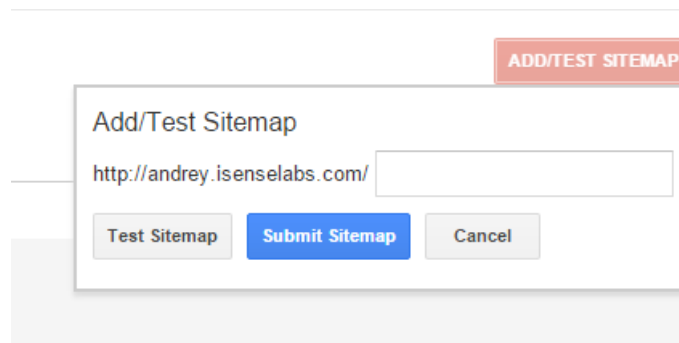
2. Go to Google Webmaster Tools  
at <https://www.google.com/webmasters/tools> .



- In order to use the Google Webmaster Tools, your website should be registered with your account. To do this, sign in to the Google Search Console, click the 'Add a site' button and follow the instructions to claim your ownership of the website.
3. Proceed to **Crawl->Sitemap** menu. This is the menu where you can see the sitemaps of your websites.



4. To add a new one select the 'Add/Test Sitemap' button and paste the Data Feed URL, you have copied from OpenCart. Remember to modify the pasted URL, leaving only the path to the sitemap page and removing the domain, as it is already predefined by Google.
5. Click '**Submit map**' and the process is finished. You can see if the sitemap is correctly created or check if there are any error messages after you refresh the *Crawl->Sitemap page*



The image shows a modal dialog box titled "ADD/TEST SITEMAP". Inside the dialog, there is a sub-header "Add/Test Sitemap" and a text input field containing the URL "http://andrey.isenselabs.com/". Below the input field, there are three buttons: "Test Sitemap", "Submit Sitemap", and "Cancel".

Congratulations, you have now setup a sitemap for your e-commerce website! A well linked website with a good sitemap would be easier to discover for search engines and this will almost certainly attract new customers to your store.

# How to set up a SSL certificate in OpenCart 2.0.x



This article will get you acquainted with the benefits of SSL (Secure Sockets Layer). You will also learn how to set up SSL for your OpenCart store. Let's go!

## What is SSL and why your website needs it?

When you visit a web page in your favorite web browser, an interesting process happens “under the hood”. Here is what it looks like in a nutshell. Your browser first connects to a web server and “asks” it to send some data. The web server responds and the data is transferred to your browser.

Depending on the kind of data which needs to be transferred, the browser and server follow a number of “rules”. We will call these rules a “protocol”. When you access a webpage, in most cases this is done via the HTTP (Hypertext Transfer Protocol) protocol. Almost all of the websites you visit are served via HTTP.

The good news is HTTP works great and it is now a worldwide standard. The bad news is that it is not secure. In other words, all of the data which is transferred between your web browser and the server is in a sense public, and it can be viewed by a third party without much effort.

You can guess what this means when your customers start entering their credit card details, usernames, emails, passwords, etc. into your HTTP-based website. Since this data is not properly protected, it can easily fall into the wrong hands. This is why it is important to secure your customers' sensitive information on as many levels as possible.

SSL (Secure Sockets Layer) is a cryptographic protocol, which is used to add an extra level of security to your HTTP requests. The resulting protocol is called HTTPS (HTTP over SSL). It does the same thing as HTTP, but with the added benefit that the transferred data is encrypted and it can be read only by the browser and the server (kind of like the Enigma machine the Nazis used in World War II).

If you are running a website which handles sensitive data (like an OpenCart e-commerce store), HTTPS is a must-have. Read on to see how to configure it.

## **Setting up SSL on OpenCart**

### **Step 0: (Preparation) Purchasing a certificate**

Before you can setup your SSL, you need to purchase an SSL certificate first. Contact your web hosting provider and ask them about this. Your SSL certificate must be installed on the hosting account on which you are running your OpenCart website.

## Step 1: Configuring OpenCart

Fortunately, OpenCart have given you a very easy way to enable SSL. Just login to your admin panel and navigate to the following page: **Admin > System > Settings > [Edit] > Server**

Now find the setting “Use SSL” and set it to “Yes”. Click the “Save” button on the top right.

## Step 2: Modifying the config.php files

We are almost done. What is left is to make sure that your OpenCart configuration files have appropriate values.

Using your favorite FTP client, download the file `/config.php` which is located in the root folder of your OpenCart website. Open it for editing.

Notice that there is a line which looks something like this:

```
define('HTTPS_SERVER', '{your-website-domain}');
```

**Make sure that** `{your-website-domain}` **starts with** `https`.

Save the file and upload it to your server.

Now download the file `/admin/config.php` and open it for editing.

Notice that there are 2 lines which look something like this:

```
define('HTTPS_SERVER', '{your-website-domain}');  
define('HTTPS_CATALOG', '{your-website-domain}');
```

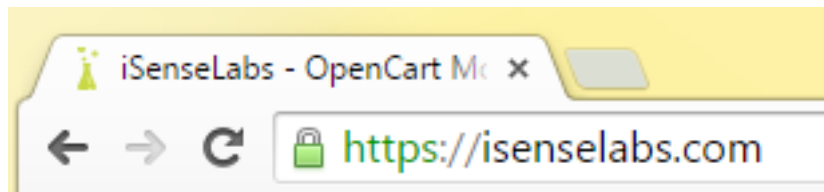
**Make sure that** `{your-website-domain}` **starts with** `https`.

Now save the file and upload it to your server. Great job!

## Step 3: Testing

Open up your website by typing `https://{your-website}` in your favorite web browser.

If everything is set up correctly, you will see a green padlock in your URL. It should look like this:



## In conclusion

Congratulations! This was not so hard, was it? If you have any questions, let us know in the comments below.

# How to Add Store Credit to Customer Accounts in OpenCart 2.0.x?



In some cases it may become necessary to add credit to your client's account, this being either a refund, a compensation, or a reward. OpenCart allows you to do this through the admin panel and this tutorial will guide you through the steps of this process.

1. Login to the admin panel of your webstore and navigate to the menu **Sales > Customer > Customer**
2. Select the customer, to whom you wish to add credit, and click the **Edit** button in the right column

Customer Name	Customer Group	Approved	Date Added
<input type="text" value="Customer Name"/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value="Date Added"/>
<input type="text" value="E-Mail"/>	<input type="text" value=""/>	<input type="text" value="IP"/>	<input type="text" value=""/>


  

<input type="checkbox"/>	Customer Name	E-Mail	Customer Group	Status	IP	Date Added	Action
<input type="checkbox"/>	Test Customer	test@mail.com	Default	Enabled	77.70.61.149	26/08/2015	<input type="checkbox"/> <input type="lock"/> <input type="edit"/> <input type="delete"/>
<input type="checkbox"/>	test test	georgiev@isenselabs.com	Default	Enabled	195.191.148.31	25/08/2015	<input type="checkbox"/> <input type="lock"/> <input type="edit"/> <input type="delete"/>

Showing 1 to 2 of 2 (1 Pages)

3. Open the Transactions tab. This is the place where you will create the credit amount
4. In the description field, write Credit, and type the necessary amount in the text box below.

Customers [Home](#) / [Customers](#)

 Edit Customer

General History **Transactions** Reward Points IP Addresses

Date Added	Description




Description

Amount

5. Click the **'Add Transaction'** button and then **Save**

Congratulations, you successfully added credit to the balance of your customer! When they create an order from your store, the credit will be deducted from the payment.

## Shopping Cart (1.00kg)

Image	Product Name	Model	Quantity	Unit Price	Total
	HP LP3065 Delivery Date: 2011-04-22 Reward Points: 300	Product 21	1  	\$122.00	\$122.00

### What would you like to do next?

Choose if you have a discount code or reward points you want to use or would like to estimate your delivery cost.

Use Coupon Code ▾

Use Gift Voucher ▾

Estimate Shipping & Taxes ▾

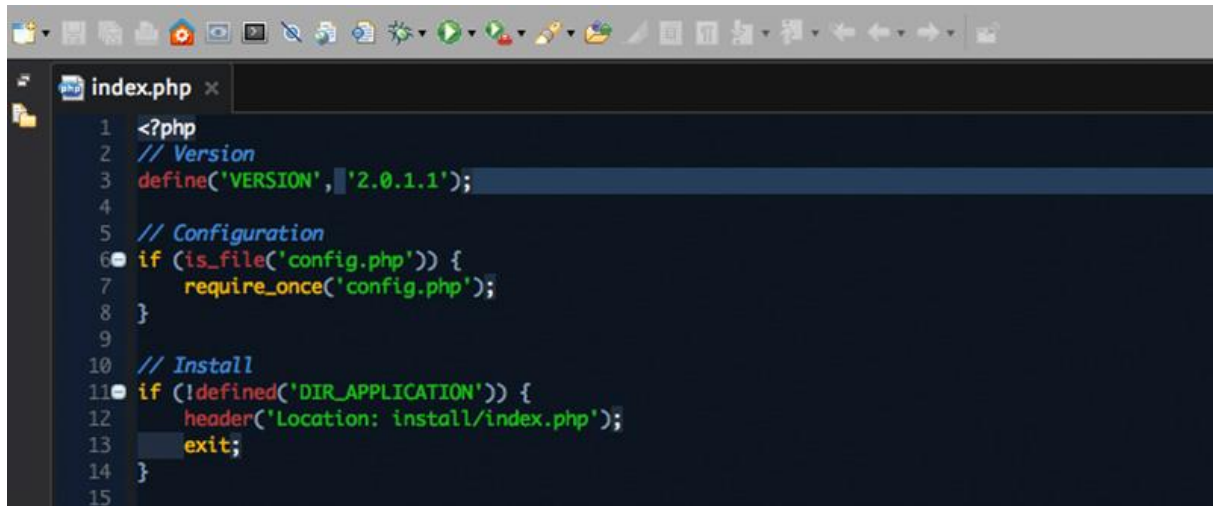
<b>Sub-Total:</b>	\$100.00
<b>Eco Tax (-2.00):</b>	\$2.00
<b>VAT (20%):</b>	\$20.00
<b>Store Credit:</b>	\$-122.00
<b>Total:</b>	\$0.00

[Continue Shopping](#)

[Checkout](#)



# How to find out which OpenCart version am I using



```
1 <?php
2 // Version
3 define('VERSION', '2.0.1.1');
4
5 // Configuration
6 if (is_file('config.php')) {
7     require_once('config.php');
8 }
9
10 // Install
11 if (!defined('DIR_APPLICATION')) {
12     header('Location: install/index.php');
13     exit;
14 }
15
```

Have you ever thought about making any changes in your OpenCart store, install new theme or modules? If so, before starting to look for a module or a theme, you need to know the specifications of your own store in order to find the compatible extensions you may need.

In this short article we will show you the first step in recognizing your store specifications, knowing the OpenCart version you are running.

Currently, there are 13 [OpenCart](#) versions available, since the first version OpenCart 1.5.4 released on August 16, 2012. Typically, OpenCart timeline is separated in two milestones, first one includes OpenCart 1.5.x versions starting from OpenCart 1.5.4 till 1.5.6.4, and the second milestone with versions from OpenCart 2.0.0.0 till 2.0.3.1. Between two major versions there is a big difference in the admin panel interface. Therefore, the OpenCart interface itself can help you distinguish between versions 1.5.x and 2.0.x.

## OpenCart 1.5.x

Let's start with finding out the version of an OpenCart 1.5.x store by accessing the data from the store admin panel. When opening your store admin panel, if you are using the default admin theme of OpenCart, you may have the following view :

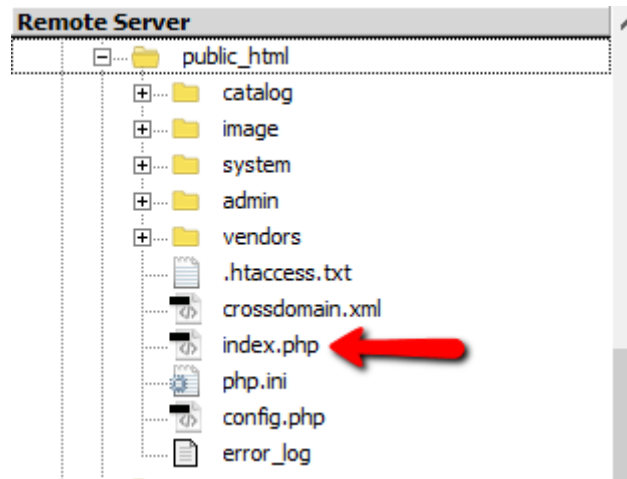


Focus on the footer of the page, and there you will see the version of this store.

In the current example, the running OpenCart version is OpenCart 1.5.6.4. If you are in Admin Panel, in the footer of each page you will see the used version of OpenCart.

Another way to check the version of your current OpenCart store, is to access this information from your `index.php` file.

Open your OpenCart installation in your server. By default, your OpenCart installation should be located in the `public_html` directory. Locate the file **`index.php`**.



Open this file and in the third line of code you will find the current OpenCart version used in your store.

```
<?php
// Version
define('VERSION', '1.5.6.4');
```

Now you can proceed with choosing the right modules and themes compatible with your OpenCart store version.

## OpenCart 2.0.x

A big difference in the admin interface of the OpenCart versions was in the transition from OpenCart 1.5.x to 2.0.x in October 1, 2014. As you can see in the picture below, different from OC 1.5.x in the login admin page of OpenCart 2.0.x the version of the OpenCart is not shown.

Please enter your login details.

**Username**

**Password**

[Forgotten Password](#)

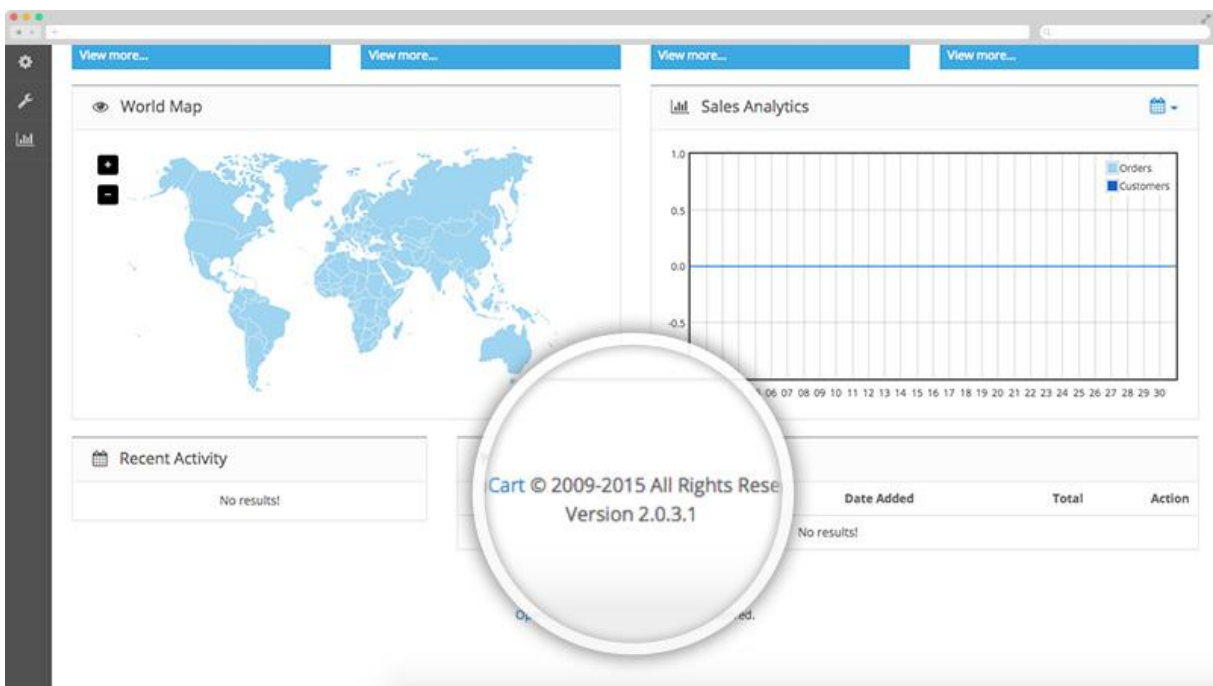
[Login](#)

OpenCart © 2009-2015 All Rights Reserved.

But how to find out which exact version is it?

As shown in the previous section for OpenCart 1.5.x, there are two ways to access this information. You can either check your store admin panel or open file **index.php**.

At first, login to your account and scroll down in the end of the page, there you will find the required information.



The exact version in the example shown in the image is OpenCart 2.0.3.1, the newest released version of OpenCart in May 29, 2015.

As explained in the previous section for OpenCart 1.5.x, you can access the same information by opening `index.php` file. For this example the OpenCart version is shown in the third line of the code below.

```
<?php
// Version
define('VERSION', '2.0.3.1');
```

Knowing your OpenCart version not only helps you choose plugins for your store easier, but it also helps you to keep your store up-to-date and bug free. When you experience a bug in your store, you can easily search for a version specific fix. Plus, you can also look if there is a newer OpenCart version and go through the changelog to check if a known bug is already fixed or any new cool feature is implemented.

# Cookie consent notification in OpenCart



## What is the Cookie law?

The Cookie Law is a privacy law directive, which was approved in May 2011 by the European Union and it effects all of the countries in the EU. The privacy law requires all website running and targeting users in countries part of the EU to ask for permission to store cookies in the browser or simply to notify the user that the website is using cookies.

You can read more about the directive on the European commission website -

[http://ec.europa.eu/ipg/basics/legal/cookies/index\\_en.htm](http://ec.europa.eu/ipg/basics/legal/cookies/index_en.htm)

## How can I make my website compliant?

This article will show you how to easily implement your own Cookie consent notification and make your website compliant with the European Union Cookie law. By default OpenCart is storing several types of cookies - Google Analytics, AddThis, the customer

PHP Session ID and the language and currency preferences. This means that you can only notify the customer that the website is using cookies without actually asking for permission to store them on the browser.

The modification can be used in all versions of OpenCart running vQmod or installed via the extension installer in OpenCart 2.0 as an OCMOD.

## OpenCart 1.5

Let's start with the older versions first - OpenCart 1.5.6.4 and below. In order to perform modifications to files in OpenCart without overwriting them directly, we will be using vQmod. Make sure you have vQmod installed and working, you can follow this quick guide here on how to install vQmod -

<https://github.com/vqmod/vqmod/wiki/Installing-vQmod-on-OpenCart>.

Open up your favourite text editor and create a new file called `cookie_consent_notification.xml` Insert the following code in the file and upload it to your `/vqmod/xml` directory.

```
<modification>
  <id>Cookie consent in OpenCart</id>
  <version>1.4.x and above</version>
  <vqmver>2.5.0</vqmver>
  <author>isenselabs.com</author>
  <file
name="catalog/view/theme/*/template/common/footer.tpl">
    <operation error="skip">
      <search
position="before"><![CDATA[</body>]]></search>
      <add><![CDATA[
```

```
<style type="text/css">
    #cookie_consent_notification,
    #cookie_consent_notification *,
    #cookie_consent_notification *:before,
    #cookie_consent_notification *:after {
        font-family: "HelveticaNeue-
Light", "Helvetica Neue Light", "Helvetica Neue", Helvetica,
Arial, "Lucida Grande", sans-serif;
        font-weight: 300;
        font-size: 16px;
        line-height: 1.6;
        -webkit-box-sizing: border-box;
        -moz-box-sizing: border-box;
        box-sizing: border-box;
    }

    #cookie_consent_notification {
        position: fixed;
        top: 0;
        left: 0;
        right: auto;
        bottom: auto;
        margin: 0;
        padding: 12px 24px;
        color: white;
        background: rgba(0,0,0,.9);
        z-index: 999999;
        width: 100%;
        text-align: center;
    }

```



```

        -webkit-transition: all ease-out
0.3s;

        -moz-transition: all ease-out
0.3s;

        -ms-transition: all ease-out
0.3s;

        -o-transition: all ease-out 0.3s;
transition: all ease-out 0.3s;
        -webkit-transform:
translate3d(0,-100%,0);

        -moz-transform: translate3d(0,-
100%,0);

        -ms-transform: translate3d(0,-
100%,0);

        -o-transform: translate3d(0,-
100%,0);

        transform: translate3d(0,-
100%,0);

        -webkit-backface-visibility:
hidden; /* Chrome, Safari, Opera */
        backface-visibility: hidden;
    }

    #cookie_consent_notification.active {
        -webkit-transform:
translate3d(0,0,0);

        -moz-transform:
translate3d(0,0,0);

        -ms-transform:
translate3d(0,0,0);

        -o-transform: translate3d(0,0,0);
transform: translate3d(0,0,0);
    }

```

```

    }

    #cookie_consent_notification
.disable_cookie_consent_notification {
        display: inline-block;
        text-decoration: none;
        text-align: center;
        font-weight: bold;
        margin: 0 10px;
        padding: 0 6px;
        color: #32C0F4;
    }
</style>
<script type="text/javascript">
    (function($) {
        (function(g,f){"use strict";var
h=function(e){if("object"!==typeof e.document)throw
Error("Cookies.js requires a `window` with a `document`
object");var b=function(a,d,c){return
1===arguments.length?b.get(a):b.set(a,d,c)};b._document=e.docu
ment;b._cacheKeyPrefix="cookey.";b._maxExpireDate=new
Date("Fri, 31 Dec 9999 23:59:59
UTC");b.defaults={path:"/",secure:!1};b.get=function(a){b._cac
hedDocumentCookie!==b._document.cookie&&b._renewCache();return
b._cache[b._cacheKeyPrefix+a]};b.set=function(a,d,c){c=b._getE
xtendedOptions(c);c.expires=b._getExpiresDate(d===f?-
1:c.expires);b._document.cookie=b._generateCookieString(a,d,c)
;return b};b.expire=function(a,d){return
b.set(a,f,d)};b._getExtendedOptions=function(a){return{path:a&
&a.path||b.defaults.path,domain:a&&a.domain||b.defaults.domain
,expires:a&&a.expires||b.defaults.expires,secure:a&&a.secure!=
=f?a.secure:b.defaults.secure}};b._isValidDate=function(a){ret
urn"[object
Date]"===Object.prototype.toString.call(a)&&!isNaN(a.getTime())

```

```

});b._getExpiresDate=function(a,d){d=d||new Date;
"number"===typeof a?a=Infinity===a?b._maxExpireDate:new
Date(d.getTime()+1E3*a):"string"===typeof a&&(a=new
Date(a));if(a&&!b._isValidDate(a))throw Error("`expires`
parameter cannot be converted to a valid Date
instance");return
a};b._generateCookieString=function(a,b,c){a=a.replace(/[\#$%+
\^`|]/g,encodeURIComponent);a=a.replace(/\/g,"%28").replace(/
\/g,"%29");b=(b+"").replace(/[\!#$%&+\\--:<-\[\]-
~]/g,encodeURIComponent);c=c||{};a=a+"="+b+(c.path?"path="+c.
path:"");a+=c.domain?"domain="+c.domain:
"";a+=c.expires?"expires="+c.expires.toUTCString():"";return
a+=c.secure?"secure":""};b._getCacheFromString=function(a){va
r d={};a=a?a.split("; "):[];for(var c=0;c<a.length;c++){var
e=b._getKeyValuePairFromCookieString(a[c]);d[b._cacheKeyPrefix
+e.key]===f&&(d[b._cacheKeyPrefix+e.key]=e.value)}return
d};b._getKeyValuePairFromCookieString=function(a){var
b=a.indexOf("="),b=0>b?a.length:b;return{key:decodeURIComponen
t(a.substr(0,b)),value:decodeURIComponent(a.substr(b+1))}};b._
renewCache=function(){b._cache=
b._getCacheFromString(b._document.cookie);b._cachedDocumentCoo
kie=b._document.cookie};b._areEnabled=function(){var
a="1"===b.set("cookies.js",1).get("cookies.js");b.expire("cook
ies.js");return a};b.enabled=b._areEnabled();return
b},e="object"===typeof g.document?h(g):h;"function"===typeof
define&&define.amd?define(function(){return
e}):"object"===typeof exports?("object"===typeof
module&&"object"===typeof
module.exports&&(exports=module.exports=e),exports.Cookies=e):
g.Cookies=e)("undefined"===typeof window?this>window);

```

```

$(document).ready(function($) {
    var show_notification = true;

    var html = '<div
id="cookie_consent_notification">';

```

```

        html += 'This website
uses cookies to improve your user experience.';

        html += '<a
href="javascript:void(0);"
class="disable_cookie_consent_notification">';

        html += 'Continue';

        html += '</a></div>';

        if (typeof Cookies == "undefined"
|| typeof Cookies != "function") {

            show_notification = false;

        } else if (typeof
Cookies.get('disable_cookie_consent_notification') !=
"undefined" &&
Cookies.get('disable_cookie_consent_notification') == 'true')
{

            show_notification = false;

        }

        if (show_notification) {

            $('body').append(html);

            $('#cookie_consent_notification').addClass('active');

            $('.disable_cookie_consent_notification').on('click',
function(event) {

                event.preventDefault();

                $('#cookie_consent_notification').removeClass('active');

                Cookies.set('disable_cookie_consent_notification',
'true');

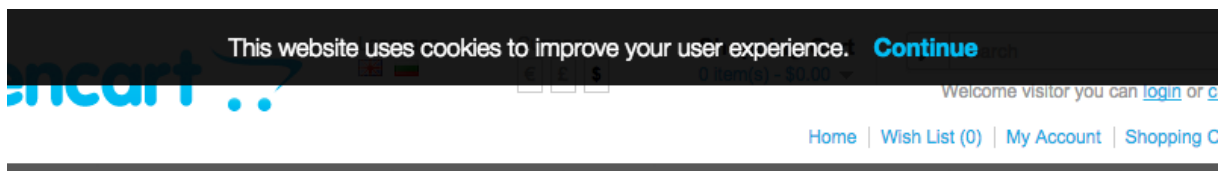
```

```

                });
            }
        });
    })(jQuery);
</script>
]]></add>
</operation>
</file>
</modification>

```

The above modification will add a notification bar to your website, which looks like this:



## OpenCart 2.0

For stores running OpenCart 2.0 use **the `cookie_consent_notification.ocmod.xml` OCMOD**, which you can download from our public repository at GitHub here <https://github.com/iSenseLabs/tutorials/tree/master/cookie-consent-notification-in-opencart>.

After download the extension file you will need to install it using the OpenCart Extension Installer.

1. Open your OpenCart administration.
2. Navigate to **Extensions -> Extension Installer**.
3. Click on the Upload button and locate the downloaded **`cookie_consent_notification.ocmod.xml`** file.

4. OpenCart will install it automatically and you will get a success message.
5. Go to **Extensions** -> **Modifications** and click on the [Refresh] button.
6. That's it!

Thank you for reading and hope that we've helped with this little tutorial!

The full source code of this tutorial can be found at our public GitHub repository: [Download Source](#)

# Customizing your store maintenance mode tutorial

Our web developers are currently performing magical spells to enhance your user experience. We will be back in a couple of hours. Thank you for your patience.



While waiting you might be interested in browsing our products in [OpenCart](#)

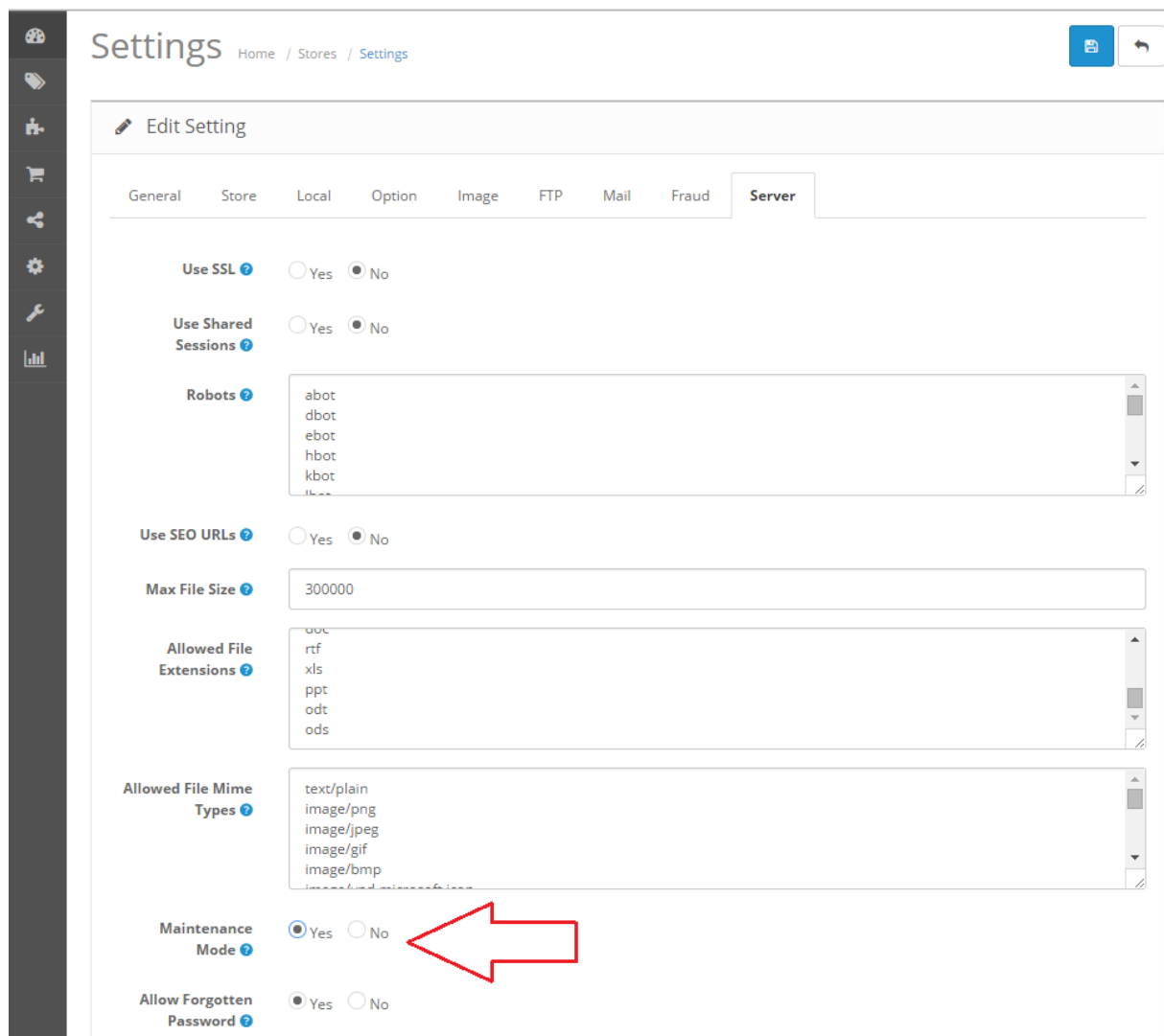
As an OpenCart store owner you are most probably aware of the Maintenance mode functionality embedded in OpenCart. The maintenance mode allows you to ‘turn off’ your website for your customers while it’s being worked on. When the mode is on, your customers will see a page that informs them that your website is down due to a scheduled or unscheduled maintenance. Enabling this mode is the easy and smart way of preventing your customers from the negative experience of encountering errors and warnings when trying to access your website.

The maintenance mode page itself means that the website is currently not working. However, it is always nicer for the customers to see a maintenance mode that catches their attention and makes them want to come back. In this blog post we will show you how you can achieve that in your OpenCart store by following the following three steps:

1. Enable your Maintenance Mode
2. Change the text in your Maintenance Mode
3. Customize your Maintenance Mode page

## Enable your Maintenance Mode

We shall start with enabling the Maintenance Mode. Login to the admin panel of your website and navigate to Settings=>Store. Choose the store in which you want to enable the mode and click Edit. Go to the Server Tab and find the Option: Maintenance Mode. Check the Yes option and Save the changes.

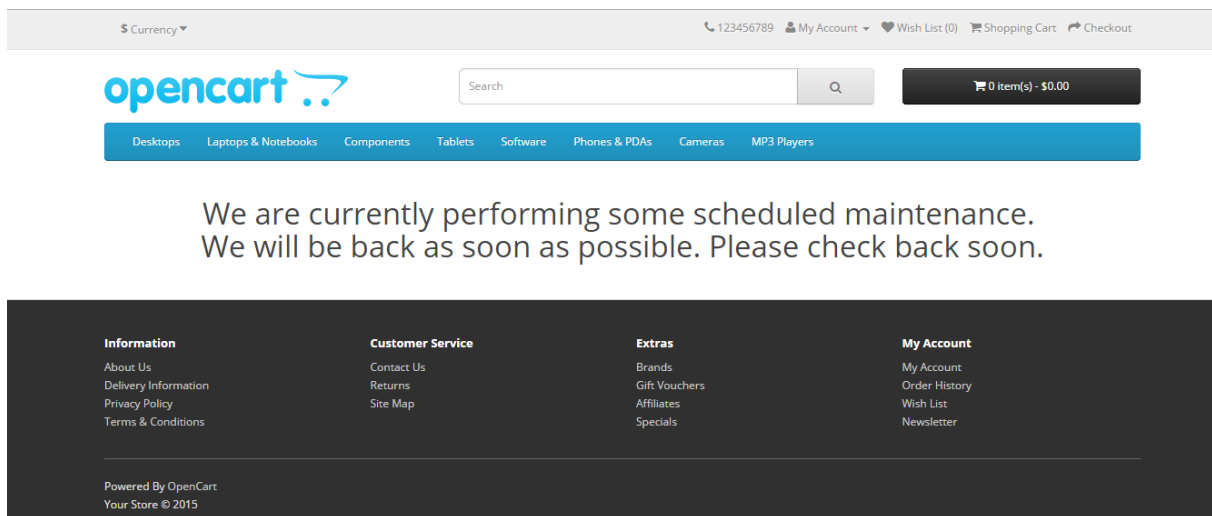


The screenshot shows the 'Settings' page in an admin panel. The breadcrumb navigation is 'Home / Stores / Settings'. The page title is 'Settings' and the current tab is 'Server'. The 'Edit Setting' section is active. The 'Maintenance Mode' option is checked to 'Yes', indicated by a red arrow pointing to the 'Yes' radio button. Other options include 'Use SSL', 'Use Shared Sessions', 'Robots', 'Use SEO URLs', 'Max File Size', 'Allowed File Extensions', and 'Allowed File Mime Types'.



Note that if you are logged in to your admin panel and you access your store, you will see it with the maintenance mode turned off. You can either start a new session and clear your cookies, or you can open up your website in an Incognito Window if you want to check how the other users will see your website while in maintenance mode.

The default Maintenance Mode in OpenCart 2.0.x should look something similar to this one:



## Change the text in your Maintenance Mode

In order to change the message of your maintenance mode, you will need to modify the language file of the maintenance mode page. To do so, you will have to locate the maintenance.php file in the following location on your server:

admin/catalog/language/english/common/maintenance.php. You can either use a FTP client such as FileZilla or Dreamweaver, or you can access the file through your Cpanel=>File Manager and then edit the file by right click on it => Edit. In both cases you will see the following code:

```
<?php
// Heading
```

```
$_['heading_title']    = 'Maintenance';  
// Text  
$_['text_maintenance'] = 'Maintenance';  
$_['text_message']     = '<h1 style="text-align:center;"><span  
style="background-color:Yellow;">The website is currently down  
for maintenance. </span><br/><span style="background-  
color:Yellow;">We expect to be back in a couple of hours.  
Thank you for your patience.</span></h1>';  
?>
```

The highlighted text is the one that can be replaced with whatever you wish. After modifying the file make sure to save it and upload it to your server.

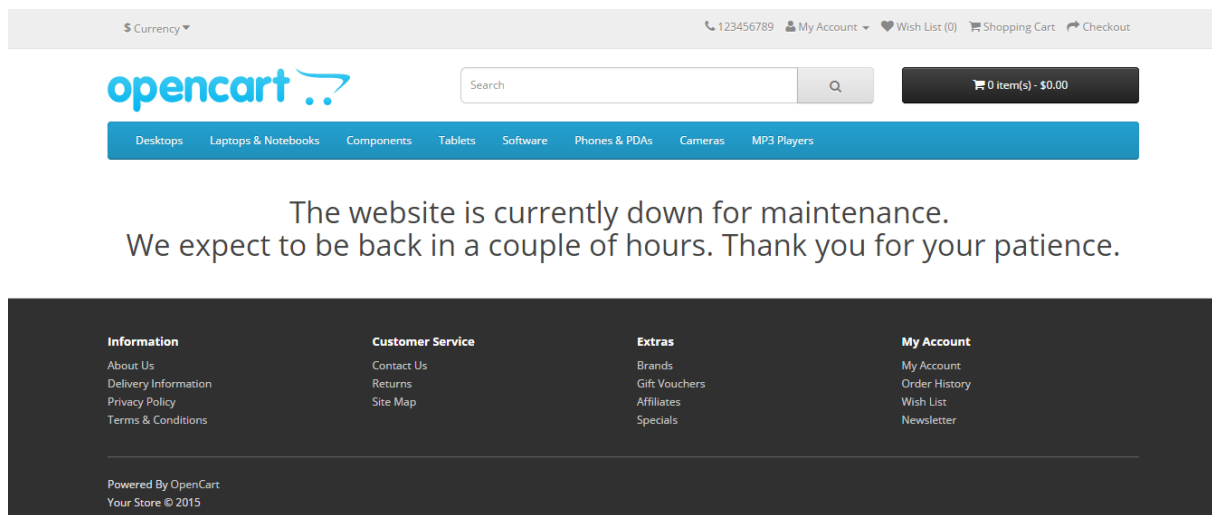
**\*Note** that the following file path is for a webstore that uses English language. If your store uses another language or is multi-store, you will need to modify the file in the following location:

admin/catalog/language/your-language/common/maintenance.php

**\*\*Note** that the text message is encapsulated in title tags: <h1></h1>, it is centrally aligned: style="text-align:center;" and there is a new line after the first sentence: <br/>. These elements are known as HTML text formatting elements. You can find more information about them and how to use them

here: [http://www.w3schools.com/html/html\\_formatting.asp](http://www.w3schools.com/html/html_formatting.asp)

The end result of the changes in the text should be something similar to this one:



## Customize your Maintenance Mode page

Apart from just changing the text of your maintenance mode you can do some further modifications that will make the page even cooler. You can add images and styles to your page so that it can grab the attention of your customers and make them want to come back. I suppose that most of you have come across some catchy Maintenance Mode designs and in the following paragraph we will show you how you can achieve this.

1. You need to upload the images that you would like to use in your Maintenance Page in the following folder: *catalog/view/theme/default/image*
2. After that you need to locate the following file: *catalog/view/theme/default/template/common/maintenance.tpl* and open it for edit.
3. Edit the template file:
  - You will notice that the template file contains the `$header` and `$footer` variables. They are used to display your OpenCart header and footer. The other variable is the `$message`, which you have already changed in the language file

- If you want to add the image that you have uploaded to your server you will need to add the following script wherever you would like the image to be displayed:

```

```

Where src is the name of the image with the path to it and the width and the height of the image set its dimensions. If you have more than one image just add the code for each image:

```

```

```

```

```

```

- You can also set different styles to the div containers by using the style="" element as shown in the example below.

```
<div class="container" style="background-color:
rgb(246,246,246); text-align:center;">
```

For more information you can check out the following link:[http://www.w3schools.com/html/html\\_css.asp](http://www.w3schools.com/html/html_css.asp)

- You can also add some other custom text and hyperlinks as shown in the example below:

```
<h3> While waiting you might be interested in browsing our
products in <a
href="http://www.opencart.com/index.php?route=extension/extens
ion&filter_username=iSense">OpenCart</a></h3>
```

Where the <a></a> tags encapsulate the hyperlink and in the href parameter you should put the URL of you link.

Here is the example code that we generated for the maintenance.tpl page is the following:

```

<?php echo $header; ?>

<div class="container" style="background-color:
rgb(246,246,246); text-align:center;">

    <div class="row">

        <div class="col-12" style="background-color: #212325;
color: #FFF; font-family:Arial Narrow; padding:50px
20px;"><?php echo $message; ?></div>

            <h3> While waiting you might be interested in browsing our
products in

                <a
href="http://www.opencart.com/index.php?route=extension/extens
ion&filter_username=iSense">OpenCart</a>

            </h3>

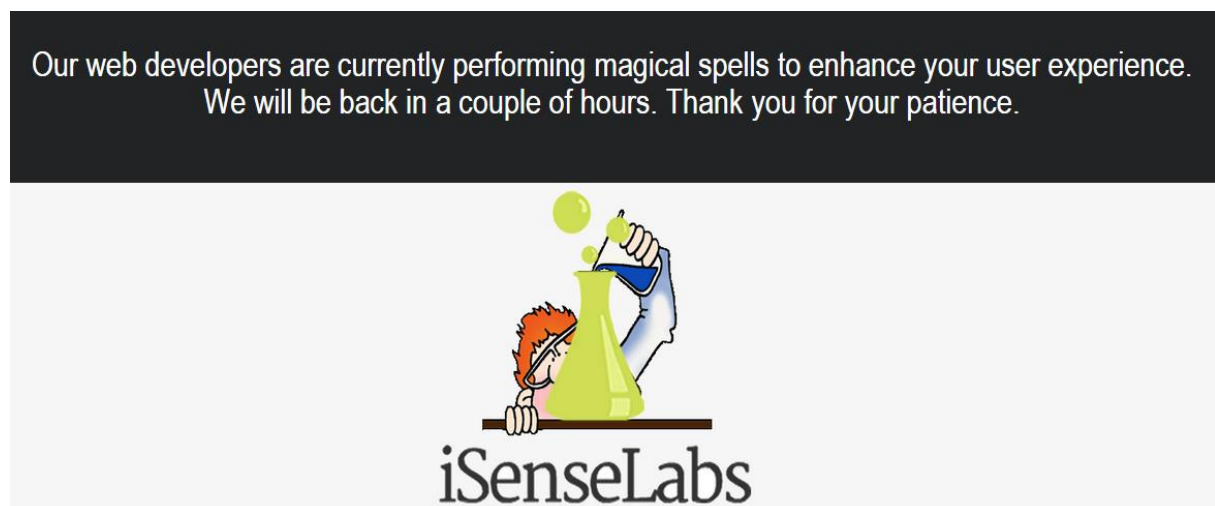
        </div>

    </div>

<?php echo $footer; ?>

```

It outputs the following maintenance page.



We hope that you will find this tutorial helpful and that you will think twice about the appearance of your maintenance mode and the impression it leaves on your customers. As a conclusion we are

suggesting you to check some creative and inspiring maintenance modes examples, which would not leave you indifferent towards these companies and their websites:

Twitter:



Digg:



---

## We'll be back shortly.

---

Digg will be down for a brief period, while we make some changes.

While not digging, the digg crew recommends:

Adam <a href="#">LivePlasma</a>	Amar <a href="#">Grow-a-Brain</a>
Anton <a href="#">Escher and the Droste effect</a>	Aubrey <a href="#">More crappy children's art work</a>
Beth <a href="#">TechCrunch</a>	Bill <a href="#">Spoon</a>
Bob <a href="#">Surfline</a>	Brian L. <a href="#">Rush</a>
Bryan W. <a href="#">khaaan!</a>	Chris <a href="#">Blueshirt Bulletin</a>
Dan <a href="#">Black Crowes</a>	Daniel <a href="#">Top Left Pixel</a>
Diane <a href="#">Mark Morford Archive</a>	Dwayne <a href="#">The Final Countdown</a>
Eli <a href="#">GeeksOn</a>	Ian <a href="#">ampedOut</a>
Jay <a href="#">The Onion</a>	Jeremy <a href="#">We Can Solve It</a>
Jim B. <a href="#">yelp</a>	Jim H. <a href="#">allblacks</a>
Joe <a href="#">moonkahn.org</a>	John M. <a href="#">paidContent.org</a>
John Q. <a href="#">sfbike.org</a>	Jordan <a href="#">Zombo</a>
Kathleen <a href="#">Yahoo!</a>	Kevin <a href="#">Pownce</a>
Kurt <a href="#">GroupLens Bloo</a>	Maoie <a href="#">The Sartorialist</a>

StumbleUpon:



37signals:

## 37signals System Status

Basecamp, Backpack, Highrise, Campfire, Writeboard, Ta-da List, and our blogs are all offline.

**UPDATE: The load balancer has been swapped and is currently being configured. We should be in the home stretch now. Again, we're incredibly sorry for this disruption. This is not how Fridays are supposed to be.**

— 11:18am CST (17:18 GMT) on January 18, 2008

**UPDATE: The technicians at our service provider are still working on installing the new load balancer. We're breathing down their neck as heavily as we can. And we profusely apologize for this unacceptable interruption of service.**

— 10:56am CST (16:56 GMT) on January 18, 2008

**UPDATE: We have located the problem to be with the load balancer setup. A new unit is being installed. We should be back shortly. Again, we're terribly sorry for this disruption of service.**

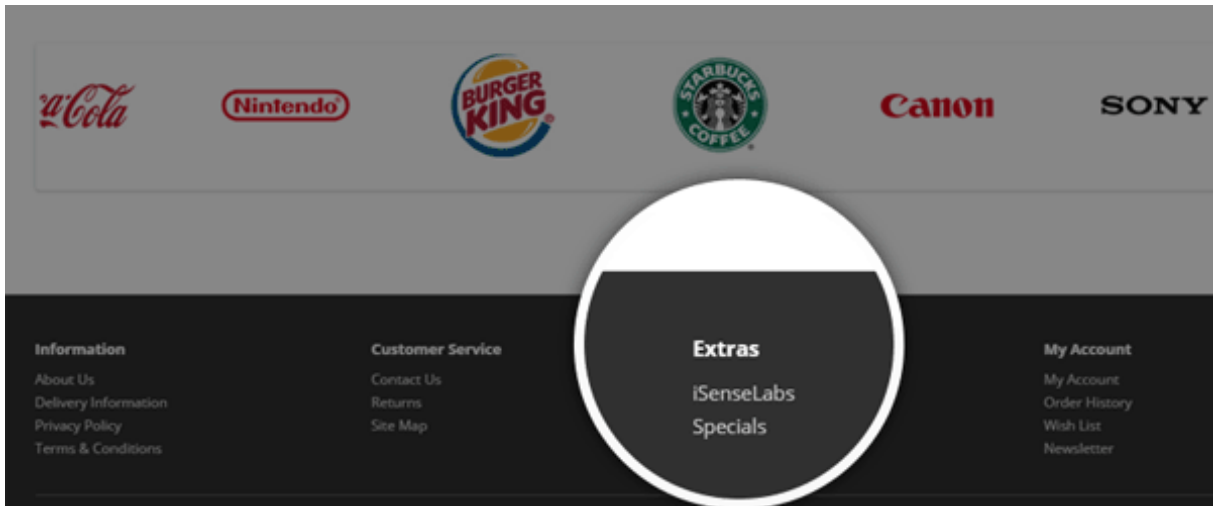
— 10:28am CST (16:28 GMT) on January 18, 2008

**All systems are currently offline as we're experiencing network outage from our provider. We're working on it right now. No data has been lost, all our machines are still working, but they're not accessible from the internet. Sorry for the inconvenience.**

— 10:03am CST (16:03 GMT) on January 18, 2008



# How to edit the OpenCart footer links



There are a few basic things regarding the OpenCart shopping cart system which every OC store owner is good to be aware of. In this blogpost we will cover the topic on the default footer in OpenCart and how you can easily edit the links in it. So keep on reading if you would like to learn how to:

1. Edit existing links
2. Delete links
3. Add new links
4. Bonus: Remove the 'Powered By' message

*Note: Please, be aware that in this tutorial we will be modifying core OpenCart files and it is highly recommended to make a backup before proceeding.*

## Edit existing links

We will start our tutorial with the editing of existing links. The first thing that we will cover would be how to change only the names of the links.

### Edit the names of the links

In order to change the names of the links in your footer, you will need to locate the following file in your server: **catalog/language/the-language-that-you-are-using/common/footer.php**. Open the file and go through the names of the links and change the desired ones in the following manner:

let's suppose you would like the **Information** link to be renamed to **Details**, then you'll need to find the following line:

```
$_['text_information'] = 'Information';
```

And change it to:

```
$_['text_information'] = 'Details';
```

After you have made the desired changes, save and upload the file to your server.

### Edit the destination of the links

If you would like to edit the destination of your links, you will need to locate the following file: **catalog/controller/common/footer.php**. The links are defined in the following way:

```
$data['special'] = $this->url->link('product/special');  
$data['account'] = $this->url->link('account/account', '',  
'SSL');
```

Let's suppose that for some reason you would like the voucher link to redirect to your shopping cart. In order to do this, you will need to modify the following line:

```
$data['voucher'] = $this->url->link('account/voucher', '', 'SSL');
```

To

```
$data['voucher'] = $this->url->link('checkout/cart', '', 'SSL');
```

This pretty much covers everything that you need to know about the modification of the existing OpenCart footer links and now we will show you how you can remove them.

## Delete links

The easiest way to remove a link from your footer, is to go to **catalog/view/theme/the-theme-that-you-are-using/template/common/footer.tpl**.

In the general case the links are displayed in the following way

```
<?php if ($informations) { ?>
    <div class="col-sm-3">
        <h5><?php echo $text_information; ?></h5>
        <ul class="list-unstyled">
            <?php foreach ($informations as $information) { ?>
                <li><a href="<?php echo $information['href'];
?>"><?php echo $information['title']; ?></a></li>
            <?php } ?>
        </ul>
    </div>
<?php } ?>
<div class="col-sm-3">
```

```

    <h5><?php echo $text_service; ?></h5>
    <ul class="list-unstyled">
        <li><a href="<?php echo $contact; ?>"><?php echo
$text_contact; ?></a></li>
        <li><a href="<?php echo $return; ?>"><?php echo
$text_return; ?></a></li>
        <li><a href="<?php echo $sitemap; ?>"><?php echo
$text_sitemap; ?></a></li>
    </ul>
</div>
<div class="col-sm-3">
    <h5><?php echo $text_extra; ?></h5>
    <ul class="list-unstyled">
        <li><a href="<?php echo $manufacturer; ?>"><?php
echo $text_manufacturer; ?></a></li>
        <li><a href="<?php echo $voucher; ?>"><?php echo
$text_voucher; ?></a></li>
        <li><a href="<?php echo $affiliate; ?>"><?php echo
$text_affiliate; ?></a></li>
        <li><a href="<?php echo $special; ?>"><?php echo
$text_special; ?></a></li>
    </ul>
</div>
<div class="col-sm-3">
    <h5><?php echo $text_account; ?></h5>
    <ul class="list-unstyled">
        <li><a href="<?php echo $account; ?>"><?php echo
$text_account; ?></a></li>
        <li><a href="<?php echo $order; ?>"><?php echo
$text_order; ?></a></li>

```

```

        <li><a href="<?php echo $wishlist; ?>"><?php echo
$text_wishlist; ?></a></li>

        <li><a href="<?php echo $newsletter; ?>"><?php echo
$text_newsletter; ?></a></li>

    </ul>

</div>

```

And in order to delete any of the links under each column, you can either delete or put the code in [html quotes](#) `<!-- code here -->`.

### Example:

Let's suppose that you would like to delete the **Wishlist** link under the **Account** column. You can either delete the row:

```

<div class="col-sm-3">
    <h5><?php echo $text_account; ?></h5>
    <ul class="list-unstyled">
        <li><a href="<?php echo $account; ?>"><?php echo
$text_account; ?></a></li>
        <li><a href="<?php echo $order; ?>"><?php echo
$text_order; ?></a></li>
        <li><a href="<?php echo $wishlist; ?>"><?php echo
$text_wishlist; ?></a></li>
        <li><a href="<?php echo $newsletter; ?>"><?php echo
$text_newsletter; ?></a></li>
    </ul>
</div>

```

Or put it in **quotes**:

```

<div class="col-sm-3">
    <h5><?php echo $text_account; ?></h5>
    <ul class="list-unstyled">

```

```

    <li><a href="<?php echo $account; ?>"><?php echo
$text_account; ?></a></li>

    <li><a href="<?php echo $order; ?>"><?php echo
$text_order; ?></a></li>

    <!-- <li><a href="<?php echo $wishlist; ?>"><?php echo
$text_wishlist; ?></a></li> -->

    <li><a href="<?php echo $newsletter; ?>"><?php echo
$text_newsletter; ?></a></li>

</ul>
</div>

```

If you would like to delete the whole column, which in our case is **Account**, you should either delete the whole container:

```

<strike>
</strike><div class="col-sm-3">
    <h5><?php echo $text_account; ?></h5>
    <ul class="list-unstyled">
        <li><a href="<?php echo $account; ?>"><?php echo
$text_account; ?></a></li>
        <li><a href="<?php echo $order; ?>"><?php echo
$text_order; ?></a></li>
        <li><a href="<?php echo $wishlist; ?>"><?php echo
$text_wishlist; ?></a></li>
        <li><a href="<?php echo $newsletter; ?>"><?php echo
$text_newsletter; ?></a></li>
    </ul>
</div>

```

Or put it in **quotes**:

```

<!-- <div class="col-sm-3">
    <h5><?php echo $text_account; ?></h5>
    <ul class="list-unstyled">

```

```

        <li><a href="<?php echo $account; ?>"><?php echo
$text_account; ?></a></li>

        <li><a href="<?php echo $order; ?>"><?php echo
$text_order; ?></a></li>

        <li><a href="<?php echo $wishlist; ?>"><?php echo
$text_wishlist; ?></a></li>

        <li><a href="<?php echo $newsletter; ?>"><?php echo
$text_newsletter; ?></a></li>

    </ul>

</div> -->

```

## Add new links

Adding new links is a little bit more complex, as it involves modifications of three files: the language file, the file in the controller and the file in the view folder. It should be done as follows:

### Example:

Let's suppose that you would like to add a link to the cart under the **Extras** column named **'iSenseLabs'**. First, you'll need to go to the language file: **catalog/language/the-language-that-you-are-using/common/footer.php**

And add the following line:

```
$_[ 'text_isenselabs' ] = 'iSenseLabs';
```

after

```
$_[ 'text_newsletter' ] = 'Newsletter';
```

Save and upload the file.

Then go to **catalog/controller/common/footer.php** and add the following line:

```
$data['text_isenselabs'] = $this->language->get('text_isenselabs');
```

### after

```
$data['text_newsletter'] = $this->language->get('text_newsletter');
```

Now, we have to define the page to which the link will redirect. If you would the link to lead to an existing page in your website, let's say the shopping cart, you should add the following line:

```
$data['isenselabs'] = $this->url->link('checkout/cart', '', 'SSL');
```

However, if you would like the link to point to another website, you should define the link like:

```
$data['isenselabs'] = 'https://isenselabs.com/';
```

The line should be added after:

```
$data['newsletter'] = $this->url->link('account/newsletter', '', 'SSL');
```

Save and upload the file.

Then go to the tpl: **catalog/view/theme/the-theme-that-you-are-using/template/common/footer.tpl** and add the following line:

```
<li><a href="<?php echo $isenselabs; ?>"><?php echo $text_isenselabs; ?></a></li>
```

### before

```
<li><a href="<?php echo $manufacturer; ?>"><?php echo $text_manufacturer; ?></a></li>
```

Save and upload the file to your server.

Now the new link will be added under the **Extras** column, above the **Manufacturer** link.



## How to remove “Powered by OpenCart”

The last thing that we will show you is how you can remove the ‘Powered by’ message at the bottom of the footer. As in our other examples, the message can be removed either by deleting or quoting the following line in the **catalog/view/theme/the-theme-that-you-are-using/template/common/footer.tpl**:

```
<p><?php echo $powered; ?></p>
```

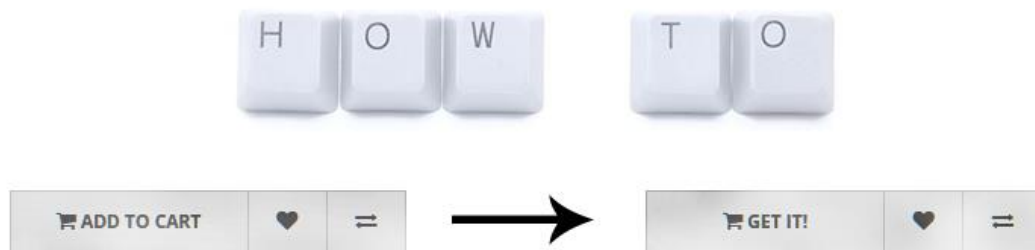
The quotation should be applied in the following manner:

```
<!-- <p><?php echo $powered; ?></p> --><span style="color:
rgb(0, 0, 0); font-family: Arial; font-size: 15px; line-
height: 1.2em; background-color: transparent;"></span>
```

## Final Words

We hope that you will find our tutorial helpful and easy-to-follow and that it will give you a basic idea on how you can easily customize the links in the footer of your OpenCart store.

# How to change the “Add to Cart” button name in OpenCart 2.0.x



*in a few easy steps*

One of the ways to make your store more unique and interesting is to change some of the text strings that are common for all online stores. This is why in this article we will show you how to change the “Add to Cart” text to something different.

## How?

Changing the button name is actually quite easy. You just have to change the language string for that phrase and you are good to go. I will show you how to do this step by step. In the following example we will change the “Add to Cart” string for the English language, but this way is applicable for all languages on your store. Lets go!

1. Access your store’s files.
2. Navigate to **catalog/language/english/english.php**.
3. Find the following string:

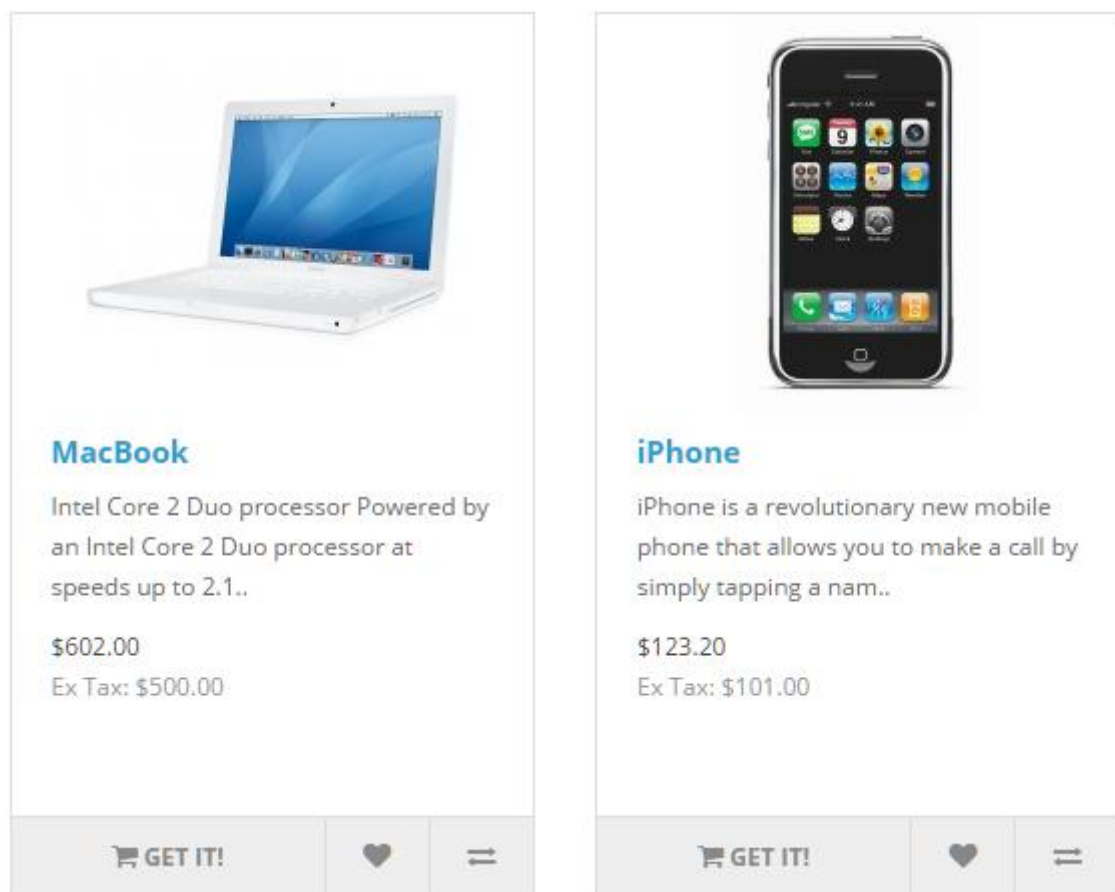
```
$_['button_cart'] = 'Add to Cart';
```

#### 4. Replace it with:

```
$_['button_cart'] = 'Get it!';
```

#### 5. That's it!

In the example above, we are changing the phrase with “Get it!”, but you can change it to whatever you want it to be. By doing this change, you will modify the string globally and it will not matter what template you are using, as long as the button text is not hardcoded in it.

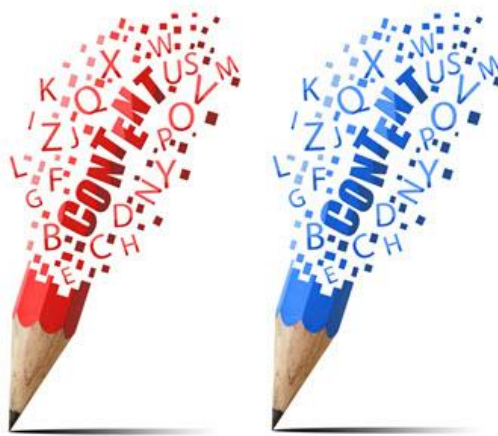


This approach was tested on **OpenCart 2.x**, but it should also work for **OpenCart 1.5.x**, because the language structure of both systems is the same.

## In Conclusion

As you can see, making this change is easy and effortless and it is not just that - you can check out all strings in that file and change not only this string, but all of them if you like. If you have any questions, let us know in the comments section below.

# How to solve the duplicate content issue in OpenCart 2.0.x



Imagine this - you are the administrator of the imaginary OpenCart store <http://myshinystore.com> and you have just set up your store to use SEO URL's.

As a result your category links look something like this:

[http://myshinystore.com/parent\\_category/child\\_category](http://myshinystore.com/parent_category/child_category) (instead of [http://myshinystore.com/index.php?route=product/category&path=11\\_3](http://myshinystore.com/index.php?route=product/category&path=11_3))

Similarly, the product links look something like this:

[http://myshinystore.com/product\\_link](http://myshinystore.com/product_link) (instead of [http://myshinystore.com/index.php?route=product/product&product\\_id=7](http://myshinystore.com/index.php?route=product/product&product_id=7))

Looks great, right? No more ugly long URLs - only meaningful paths from now on.

## So what?

Well, there is one small thing which could impact your website ranking in search engines like Google or Bing. Take for example the category link above. The same page can be accessed from:

[http://myshinystore.com/child\\_category](http://myshinystore.com/child_category) (instead of  
[http://myshinystore.com/parent\\_category/child\\_category](http://myshinystore.com/parent_category/child_category))

This is a problem because even though both links lead to absolutely the same content, the search engine will regard them as different pages on your website - the so-called “Duplicate content” issue. More information about it here:

<https://support.google.com/webmasters/answer/66359?hl=en>

## So, how do we solve this?

Since there is no setting in OpenCart to resolve this, we will need to get our hands dirty and modify a bit of code in your store. The modification will be made as an OCMOD extension to avoid changes to your core files.

**Note:** Please keep in mind that the changes we are about to make might cause conflicts with other third-party extensions on your store. If this happens, feel free to disable the modifications in order to return to the previous behavior.

**Note:** Also keep in mind that these modifications are developed for OpenCart 2.x.

### Step 1 - Prepare the file.

Without further ado, let's begin! Using your favorite text editor, create a new file called **duplicate\_url\_fix.ocmod.xml**.

There are generally two ways in which the duplicate issue can be resolved. Use only one of the approaches below, depending on your preference.

## Step 2, Approach 1 - Modify the store to use only the short versions of the links.

This will make OpenCart convert all of the SEO links to only a single word (without any paths). So as a result all links to child categories and products will look like this:

[http://myshinystore.com/child\\_category](http://myshinystore.com/child_category) (instead of  
[http://myshinystore.com/parent\\_category/child\\_category](http://myshinystore.com/parent_category/child_category))

Add the following contents to the newly created file **duplicate\_url\_fix.ocmod.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<modification>
  <name><![CDATA[Duplicate Content Fix]]></name>
  <code><![CDATA[duplicate_content_fix]]></code>
  <version><![CDATA[1.0]]></version>
  <author><![CDATA[iSenseLabs]]></author>
  <link><![CDATA[http://isenselabs.com]]></link>
  <file path="catalog/controller/common/seo_url.php">
    <operation>
      <search><![CDATA[parse_str($url_info['query'],
$data);]]></search>
      <add position="after"><![CDATA[
        $has_product_id = false;
        $has_path = false;
        foreach ($data as $query_key => $query_value)
        {
          if ($query_key == 'product_id' &&
!empty($data['route']) && $data['route'] == 'product/product')
          {
            $has_product_id = true;
          }
          if ($query_key ==
```

```

'path') {
    $has_path = true;
}
if ($has_product_id && $has_path) {
    unset($data['path']);
} else if (!$has_product_id && $has_path) {
    $path_parts = explode('_', $data['path']);
    $data['path'] =
$path_parts[count($path_parts) - 1];
}
]]</add>
</operation>
</file>
</modification>

```

## Step 2, Approach 2 - Modify the store to always use the long versions of the links.

This will make OpenCart convert all of the SEO links to the longest path possible. So as a result all links to child categories and products will look like this:

[http://myshinystore.com/parent\\_category\\_1/parent\\_category\\_2/child\\_category](http://myshinystore.com/parent_category_1/parent_category_2/child_category) (instead of [http://myshinystore.com/child\\_category](http://myshinystore.com/child_category))

Add the following contents to the newly created file **duplicate\_url\_fix.ocmod.xml**:

```

<?xml version="1.0" encoding="UTF-8"?>
<modification>
    <name><![CDATA[Duplicate Content Fix]]></name>
    <code><![CDATA[duplicate_content_fix]]></code>
    <version><![CDATA[1.0]]></version>
    <author><![CDATA[iSenseLabs]]></author>
    <link><![CDATA[http://isenselabs.com]]></link>
    <file path="catalog/controller/product/category.php">
        <operation>
            <search><![CDATA[$category_info = $this-

```



```

>model_catalog_category-
>getCategory($category_id);]]></search>
    <add position="before"><![CDATA[
        $this->session->data['last.entered.category']
= $category_id;
    ]]></add>
    </operation>
</file>
<file path="catalog/controller/common/seo_url.php">
    <operation>
        <search><![CDATA[class ControllerCommonSeoUrl
extends Controller {}]]></search>
        <add position="after"><![CDATA[
            public function findParentPath($category_id) {
                $found_path = array($category_id);

                do {
                    $category_result = $this->db-
>query("SELECT * FROM " . DB_PREFIX . "category WHERE
category_id = '" . $category_id . "'");
                    $category_id = (int)$category_result-
>row['parent_id'];

                    if ($category_id > 0 &&
!in_array($category_id, $found_path)) {
                        array_unshift($found_path,
$category_id);
                    }
                } while ($category_id != 0);

                return $found_path;
            }
        ]]></add>
    </operation>
    <operation>
        <search><![CDATA[parse_str($url_info['query'],
$data);]]></search>
        <add position="after"><![CDATA[
            $has_product_id = false;
            $has_path = false;

```

```

        foreach ($data as $query_key => $query_value)
        {
            if ($query_key == 'product_id' &&
!empty($data['route']) && $data['route'] == 'product/product')
            {
                $has_product_id = true;
            }
            if ($query_key == 'path') {
                $has_path = true;
            }
        }
        // Calculate full path
        $parent_categories_paths = array();

        if ($has_product_id) {
            // Find the true path based on the
product_id
                $parent_categories_result = $this->db-
>query("SELECT * FROM " . DB_PREFIX . "product_to_category
WHERE product_id='" . (int)$data['product_id'] . "'");

                foreach ($parent_categories_result->rows
as $parent_category) {
                    $parent_categories_paths[] = $this-
>findParentPath($parent_category['category_id']);
                }
            } else if ($has_path) {
                // Find the true path based on the last
category_id
                    $path_parts = explode('_', $data['path']);
                    $category_id =
$path_parts[count($path_parts) - 1];
                    $parent_categories_paths[] = $this-
>findParentPath($category_id);
                }
            if (!empty($parent_categories_paths)) {
                $last_entered_category = !empty($this-
>session->data['last.entered.category']) ? (int)$this-
>session->data['last.entered.category'] : 0;

```

```

                $data['path'] = implode('_',
$parent_categories_paths[0]);
                $has_path = true;
                foreach ($parent_categories_paths as
$parent_categories_path_candidate) {
                    if (in_array($last_entered_category,
$parent_categories_path_candidate)) {
                        $data['path'] = implode('_',
$parent_categories_path_candidate);
                        break;
                    }
                }
            }
        ]]></add>
    </operation>
</file>
</modification>

```

## Step 3 - Uploading the file

Almost there. Now save your file and install it with the OpenCart Extension Installer. Make sure after you upload the file to click Refresh in **Admin > Extensions > Modifications** in order for the changes to get applied.

## That's it!

Congratulations! The changes you made will help avoid the duplicate content issue. Note that this is not the only way to resolve this issue - another totally different approach would be to use canonical URL's in your pages. More information about canonical URL's can be found here:

<https://support.google.com/webmasters/answer/139066>

If you want to use canonical URL's in your website, there are a few ready modules in the OpenCart Extension store:

[http://www.opencart.com/index.php?route=extension/extension&filter\\_search=canonical](http://www.opencart.com/index.php?route=extension/extension&filter_search=canonical)

I hope you found the information above useful.

# OCMOD not working?

## OCMod not working?



DIR\_APPLICATION  
DIR\_CATALOG

Sometimes OpenCart's OCMOD system may stop working on the front-end and you may spend some quality time looking for the reason. It seems like cases like this are quite common, so here are a few guides on how to fix OCMOD related issue yourself.

## What is causing the issue?

100% of the cases we've had, the issue has been caused by the presence of the **DIR\_CATALOG** constant in the catalog's *config.php* file. Since by default this constant is only present in the admin panel, the modification function checks if it is defined and if yes, it looks for modifications in the admin directory. So when the constant is present in the catalog part, OpenCart is confused and looks for modifications in the admin, but such are usually not present there or the modified files have completely different logic and errors start appearing.

## How to fix this?

It will be best if you know which extension/customization is using the **DIR\_CATALOG** constant in the catalog and contact its developer to ask them to fix the extension and remove the constant from the *config.php* file. That way you will keep a cleaner version of

OpenCart, which helps in the future when you need to update the system.

Your second option is to modify the *system/startup.php* file. Open it and look for the modification function. It should be near the middle of the file and looks like this:

```
function modification($filename) {
    if (!defined('DIR_CATALOG')) {
        $file = DIR_MODIFICATION . 'catalog/' .
substr($filename, strlen(DIR_APPLICATION));
    } else {
        $file = DIR_MODIFICATION . 'admin/' .
substr($filename, strlen(DIR_APPLICATION));
    }

    if (substr($filename, 0, strlen(DIR_SYSTEM)) ==
DIR_SYSTEM) {
        $file = DIR_MODIFICATION . 'system/' .
substr($filename, strlen(DIR_SYSTEM));
    }

    if (is_file($file)) {
        return $file;
    }

    return $filename;
}
```

**change the following line**

```
if (!defined('DIR_CATALOG')) {
```

to

```
if (basename(DIR_APPLICATION) == 'catalog') {
```

and this should fix the issue.

# Block unwanted bots from accessing your site



All kind of bots are walking around the internet these days searching for personal information or other information which is not targeted for the general public. In this article we will show you how you can protect your site from such malicious bots.

## Prerequisites

We will be using the Project Honey Pot service, so you will need to register there and obtain your API Key. Head over to <http://www.projecthoneypot.org/> and click on the “SIGN UP FOR FREE” button. Follow the instructions and you will have your key in a minute.

## What does this service do?

Project Honey Pot is a global effort to provide us with a way to recognize malicious visitors quickly. It works based on IP address and the IP database is updated real-time. Everybody can set-up a trap for the bots and when a visitor is caught in that trap, their IP



address is being reported. This is how the information is kept up to date.

## Connect your site to Project Honey Pot

We are happy to announce that we created the [BotBlocker](#) extension for OpenCart stores running on versions 2.0.1.0 or later, which allows you to connect to Project Honey Pot in a matter of seconds. All you need to do is grab a copy of the extension and install it in your site. Then go to **Extensions > Modules > BotBlocker** enter your API Key and choose your preferred settings. You are ready to go. You can choose what type of bots you want to block which can be any of the following four options:

1. Search Engines - bots which crawl the content of your site and use the information to help people find about you
2. Suspicious - these are IP addresses which have been caught in malicious activities, but this has happened rarely
3. Harvesters - bots which crawl your site looking for email addresses
4. Comment Spammers - bots which look for form elements and use them to post malicious comments

In case you chose to block search engines, you have the option to specify which search engines exactly would you like to block. The available options here are 13:

1. Undocumented
2. AltaVista
3. Ask
4. Baidu

5. Excite
6. Google
7. Looksmart
8. Lycos
9. MSN
10. Yahoo
11. Cuil
12. InfoSeek
13. Miscellaneous

You can also set the how old should be the oldest data you want to trust. This option is set in days and the default value is 15. That means that when a particular IP is trying to access your site, if the Project Honey Pot return information older than 15 days, BotBlocker will ignore the response and will grant access to that particular IP.

On a higher level, BotBlocker shows you nice graphs about the bot traffic distribution on your site, so you can have a better visual overview of how often bots are trying to get information from your site.

There is also a list with all the IP addresses which have accessed your site, so you can manually allow or block any the known IPs. Feel free to play with BotBlocker on its demo page here:<http://botblocker.demo.isenselabs.com/admin/index.php?route=module/botblocker>

# How to ban visitors by IP address



Access control is a vital part of the management process of a website, an online store, etc. We all know that there are people on the Internet who we don't want on our website. Whether they are just annoying spammers or mischievous hackers representing a considerable threat to our website's security, it is without doubt very important to know a way or two to block them off.

## Ban by IP using the OpenCart Admin Dashboard

The Admin Dashboard provides the easiest way to enable or disable visitor access on your store. The process is fairly simple:

1. First, login to your admin control panel.
2. Go to Sales > Customers > Banned IP
3. Click the Add New button which is located in the top right corner.

4. Type in the IP address you want blocked from your website.
5. When you are ready, hit Save.

As you can see, the power to ban by IP is only a few clicks away.

## Handling security on the server side

The other way to employ access control relies on the server, rather than the OpenCart application. It can be done by using two methods - modifying the .htaccess file and using the cPanel IP Address Deny Manager. The latter, of course, is only possible if your hosting service is providing a cPanel.

### Using the .htaccess file

1. Go to your OpenCart installation.
2. Open the .htaccess file (if you don't have one, you can create one yourself, just create a new file and save it as .htaccess)
3. In this file, using the keywords Deny from, you can enlist the IPs that you want to restrict from viewing your website, eg.:  
Deny from 120.24.134.25
4. Using Allow from, you can enlist IPs which will be allowed to visit your website, eg.: Allow from 120.24.134.25
5. You can combine Deny from and Allow from using the Order keyword, eg.:

```
Order deny, allow  
Deny from all  
Allow from 120.24.134.25
```

This way you can enable only certain visitors to your website.

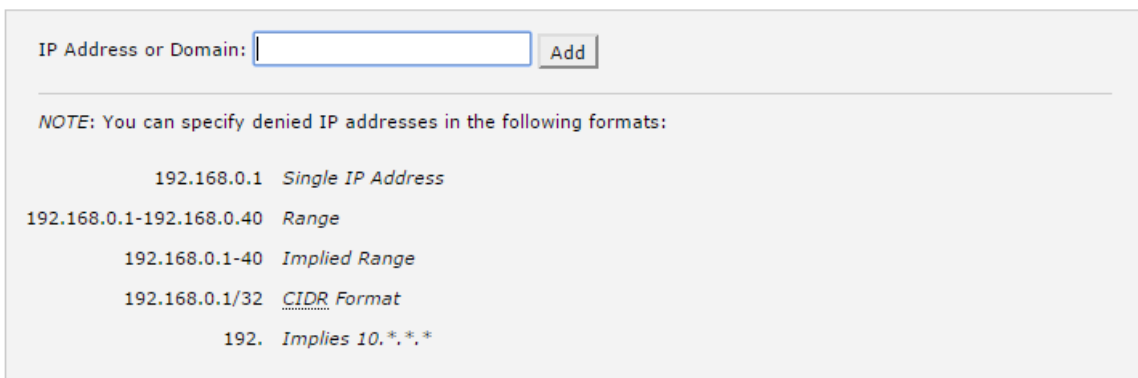
## Using the cPanel IP Address Deny Manager

1. Login to your cPanel.
2. Scroll down to the Security section.
3. Open the IP Address Deny Manager. This page you will see will be something like this:

### IP Address Deny Manager

*This feature allows you to block a range of IP addresses access to your site. If you enter an FQDN, the system will attempt to resolve it to an IP address.*

#### Add an IP address to deny:



IP Address or Domain:

---

**NOTE:** You can specify denied IP addresses in the following formats:

- 192.168.0.1 *Single IP Address*
- 192.168.0.1-192.168.0.40 *Range*
- 192.168.0.1-40 *Implied Range*
- 192.168.0.1/32 *CIDR Format*
- 192. *Implies 10.\*.\*.\**

4. Type the IP you want to block.
5. Hit Add.
6. You will be redirected to a success page, if you click Back, you can see the IP you typed in the list with other restricted IPs.

## That's all folks!

It is up to you to choose your preferred method. The one using the OpenCart admin panel is the easiest, but the others provide you with the opportunity to enlist a lot of IPs much faster. In the cPanel IP Address Deny Manager for example, you can type 120.24.134.25 - 120.24.134.50 and all the IPs in between will be added to the list. If you have any questions, please use the comment section below.

# Seven Ideas for Personalized Customer Experience in eCommerce



Welcome, John.

As in real life, most of your online visitors will visit your shop without buying anything. This would be an e-commerce store as a store front place, where anonymous people go to checkout products and fraction of them decide to buy. The other bigger part of visitors leaves without a note or even saying good bye. Imagine now that you assign a personal consultant to each one of your visitors that guides them throughout their personal motivation of being on your website. Each customer will get a tailored experience, and your conversion rate will go up dramatically, and [this Mojn data](#) backs it up. In short - one size does not fit all, focus on personalization. In this blog post I will give you a few ideas how you can personalize the experience of your shopping cart.

## Product builder

Product builder is a set of interactive steps the customer should take to personalize a product. Usually the product builder is a custom-built software specifically working for your personalizable product. Examples of shopping carts with product builders are [Casetify](#) and [SkinIt](#).

## Inline chat

Talking to your anonymous visitors on a pre-sale level cuts the distance to sale. Having a one-tap inline chat is a good idea if you can afford an agent to answer all the questions and inquiries. The two popular chat services here are [Zopim](#) and [Olark](#)

## Personal product offer

Creating personal product offers can be achieved in different ways. One of the tools you can use to do this in OpenCart would be the [LoginToSeePrice](#) free extension available. This extension together with the OpenCart native functionality would allow you to set different prices for customer groups and therefore create personal experience. Alternative to this would be "login to see your personal discount approach" where you can apply discounts to customers when they have logged in.

## Location-based content

By detecting the geo-location of the visitor you can offer specific content to the place where the visitor comes from. A popular implementation of this is redirects to local stores based on the IP address of origin of the visitor. Good example here is [Microsoft Store](#).

## Birthday greeting

Send personalized greetings to your customer for a special personal moment like a birthday. If using OpenCart you can use [BirthdayReminder](#) to automate these personalized messages.

## Personal messages

Sending personal messages, both emails and SMS contribute to the overall personal contact with your customer. Using the customer name and data if needed is a must for achieving this experience. Emails are native for almost every shopping cart platform, as for short messages you can use services like [smsbump.com](https://smsbump.com)

## Abandoned cart

This is a personal message to an user that has tried to checkout products on your site but has not made it to the end. Practical example of this behavior is the [AbandonedCarts](#) module by iSenseLabs.

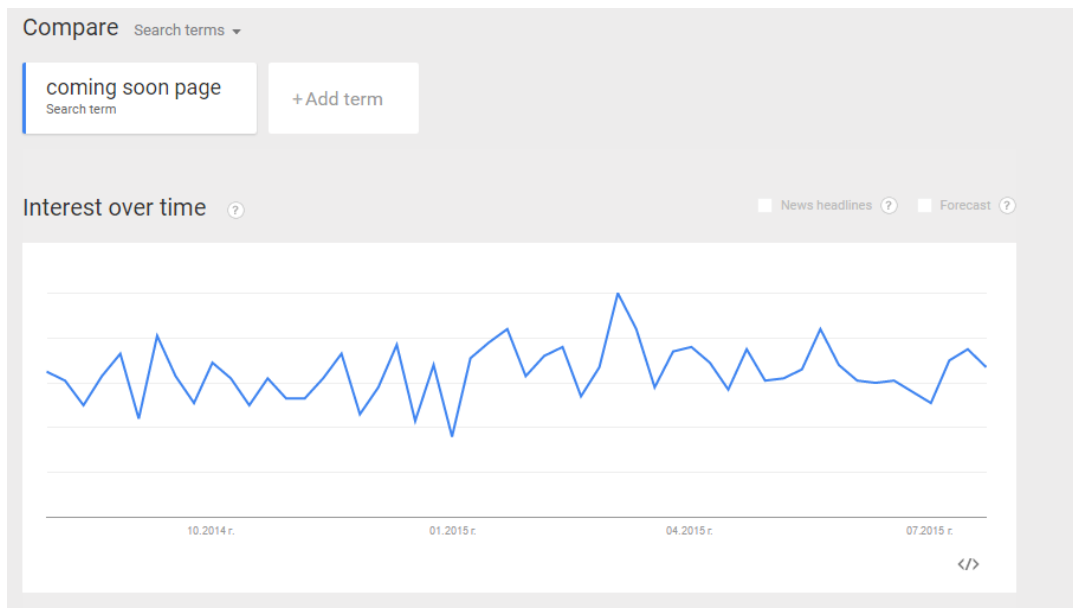


# Five beautiful examples of coming soon pages to make customers eager to visit your ready website



Coming Soon pages are well known and widely used concept in web pages design and marketing strategies. They have been around since the dawn of Internet but I doubt any of you has spent more than two seconds on them. We all know the ugly “Under construction” text accompanied by a picture of a traffic cone or something similar. A couple of years ago however, web designers saw the potential which lays behind this empty page. That’s how coming soon pages were born. A coming soon page is more often than not a one page website, whose sole purpose is to inform your audience about your upcoming or reworked project. But the way you put it to the public - well that is the key part. Creating buzz and expectation even before your website's official launch could sometimes be the difference between success and failure.

The use of a coming soon page has revealed a huge potential in marketing perspective, which has increased its value. According to Google trends the interest towards coming soon pages has increased and has kept its levels constantly high during the past one year. That is due to the fact that almost every project during this period has started with a coming soon page.



But what makes a coming soon page successful? Here is a brief list of what I think are some of the key factors for building the perfect coming soon page.

- **Simplicity** - a coming soon page has a very short lifespan. After all, it is designed just to prepare the audience for the upcoming product or service. In this relation it should be as simple and minimalistic as possible.
- **Building curiosity teaser** - think of the coming soon page as a movie trailer. Usually they reveal the most interesting part of a movie, thus making you want to watch the whole movie. Well that is what your coming soon page should do. Provide just the essential information of what is about to happen, but

do not give out everything. The curiosity is what will make your users come back once your project starts.

- Introducing your brand to the public - it is almost mandatory to use common brand attributes as logos, texts, fonts and colors, which you will be using in the official version of your product. In this way you will introduce your brand to the public and make it memorable and recognisable later when it is really needed.
- Start collecting emails - this is the best moment to start building your email list even before you need it. Use the curiosity you are going to build with this page to make people want to be amongst the first to know when your project launches.
- Use the power of social media - Social media is one of the most widely used channels for popularization these days. Put it in your favor. Give visitors the chance to share your awesome coming soon page and thus popularise your brand.

Once we have shared the common grounds on how the key aspects of a coming soon page I will show you couple of examples of coming soon pages below.

## **CSSPiffle**



CSSPiffle is a web application for designing web UIs or mobile apps. It is kind of extended more simplified photoshop, which helps generating HTML and CSS. The visual elements used in the page are enough to show you the most important features of the product. This way they provide just enough information to keep you interested of what is coming.

## ArteryInk



ArteryInk is a website currently live, which combines the passion of two artists for healthy living and beautiful art. Their under construction page not only gives enough information to get you interested but also has started building up content through its blog section. This is a clever approach, which helps search engine spiders start indexing your website even before it is launched.

**Noa+Micah**



Noa+Micah is a retailer web site for children clothing. This is a perfect example of utilizing your brand's logo into building a memorable impression which transcends into the live website.

## **HealerTree**



Enter email for early access. [Invite Me](#)

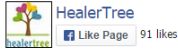
**Healers: get more clients**

Get more traffic and clients with a free listing in our search-engine optimized directory of healers.



**Clients: find the help you need**

Our goal is to be the most user-friendly, accessible, and quality directory on the web. No filler—just good people.



**Interacting with website visitors**

We'd love your input as we build.

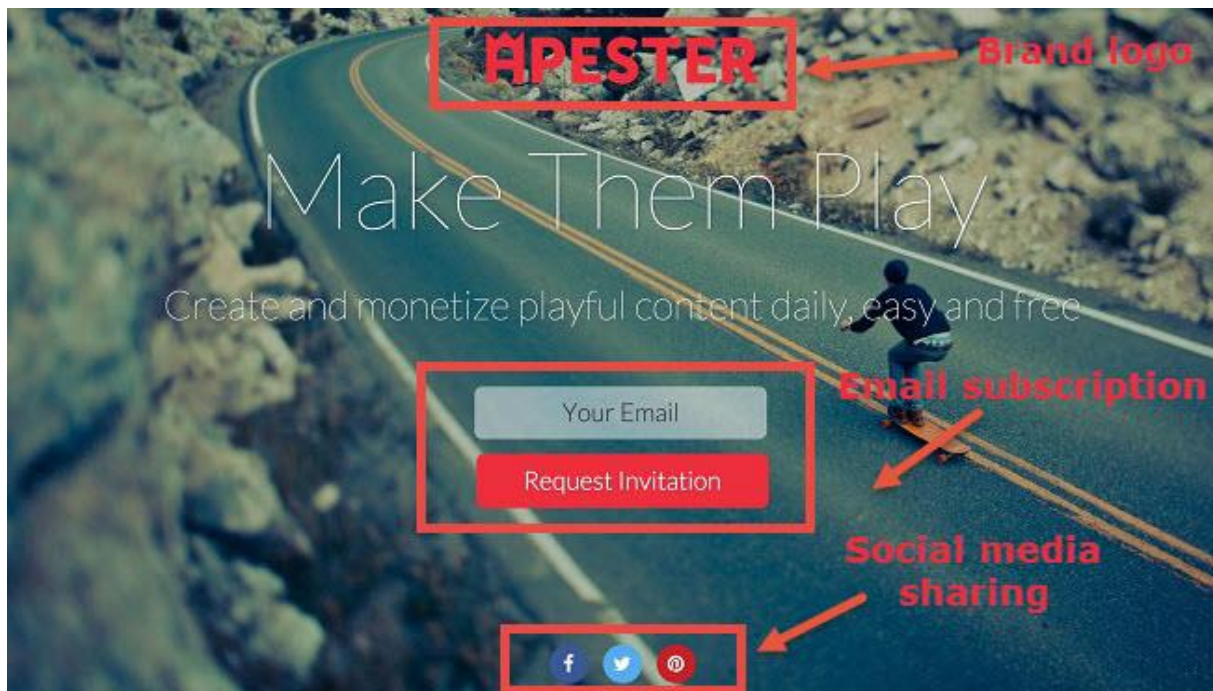
[Take the Survey](#)



LARK LABS

HealerTree.com is a website for therapists, bodyworkers, and other healers to grow their practice and do more good work. The current landing page is simple and provides enough information for you to get to know what is the site about. It also engages you by asking for a short survey, thus making you feel a part of the whole project.

**Apester**



Apester helps you create engageable content that puts the power back in your reader's' hand and maximize your websites' goals! This is a service that revolutionize the digital storytelling utilizing the voice of your readers! The coming soon page of this project is simple enough, but at the same time very attractive and intriguing. The site is already live, so you can visit it and get to know more.

Throwing in a Coming soon page is probably the best thing you can do right after purchasing your domain name. This concept will not only give you the time to develop your real project but also help you popularize it and build up your social network following list or email newsletter subscribers, but also prepare the crowd for what is coming.

If you are an OpenCart user just starting your project and you like the idea of a Coming soon page, make sure to checkout this [extension](#), which will help you build one in no time.



# The art of mastering product returns



As an online retailer, you are most probably familiar with the concept of Returned Products. Having a refund policy of any type is vital for the success and growth of your business, especially in the field of e-commerce. Ensuring your customers that they will be able to return their products, if they are not satisfied with them, for either an exchange, store credit or a complete refund, will help you build loyalty, security and reliability towards your online business. And although a liberal return policy might seem to be costly at the beginning, in the long run it is proven to be rather beneficial and effective.

Although returned products are something common in every online business and the number of stores with refund policies and not a single returned product are close to 0, there are some simple steps which will help you minimize the returns, and thus your costs.

In this blogpost we will concentrate on the most common reasons for product returns, give you some ideas on how to reduce and avoid them. Towards the end of the article you will find some

bizarre and absurd reasons for returned products which were actually used (and some of them even worked!)

## 7 most common reasons for returned products



As you might guess, one of the most common reasons for returned products is the fact that the product is not actually working. If you are a store owner of a clothing store, this is not something to worry about, but if you are selling electronics this is the paragraph for you.

Shipping a product that is malfunctioning is not something uncommon and although it brings some negative reactions in the customers it is not something totally unexpected and inexcusable. Fabric defects and faults can happen and the only way to avoid them is to increase the quality control and assurance.

In such cases the best refund practice is to propose an exchange of the defective product for a brand new one.



The clothing store owners are the ones that have encountered the wrong-size cause for returned products numerous times.

Unfortunately, in such a case there isn't much you can do apart from clearly labeling the clothing sizes. You can even put size charts and comparison tables for the different sizes worldwide.

This will be really helpful for foreign customers who are not used to your local sizing scale.

In this case the refund policy that will fit you best is to offer an item with the correct size.



Another common reason for returning products, is when the actual product doesn't match the description and/or the images used for its promotion. Unlike the previous two reasons, this one is

considered to be more unpleasant for the customer and may result in a lost credibility and negative reviews and comments from your customers. Since the feedback for an online retailer is essential for its popularity, this third cause is the one that you should try avoiding by all means. In order to do so, always make sure that the description of each product is clear and detailed enough.

Furthermore, add product images that are accurate and not misleading and in the best case scenario, taken from different angle and having the option to zoom in and out. Spending time on improving your product page will definitely result in happier and more satisfied customers.

In a case of returned products due to a non-corresponding description, the best practice is to offer either a full refund or a store credit to your customers, so that they don't leave your online store with a negative impression and frustration.



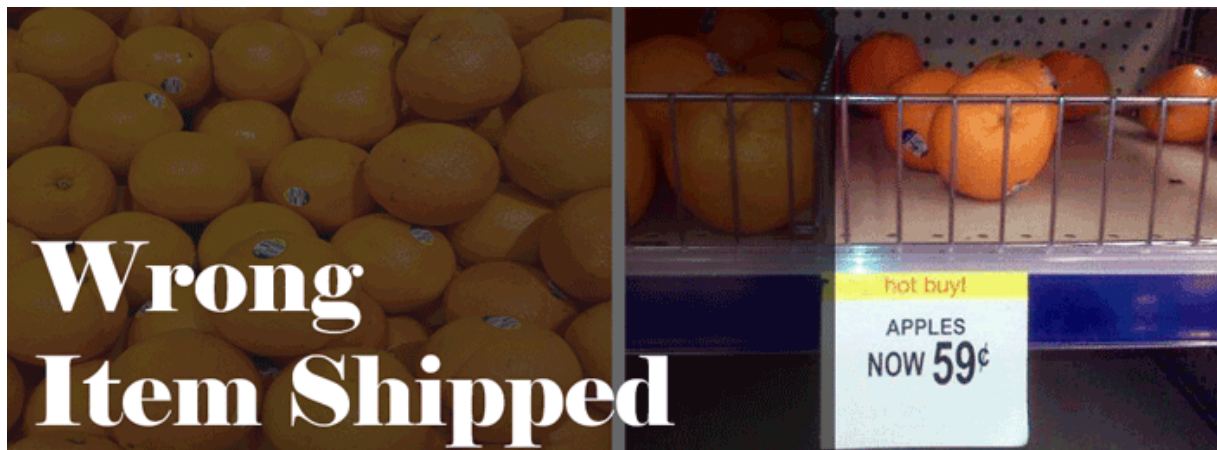
This might sound bizarre, but you'll be surprised how many people are getting distracted while shopping and ending up checked-out and paid for products which they don't want and have accidentally put in their shopping cart. In such a case you can't do much apart from asking for a confirmation before checking out and hoping that it wouldn't go unnoticed.

The refund policy that best fits this situation would be a store credit.



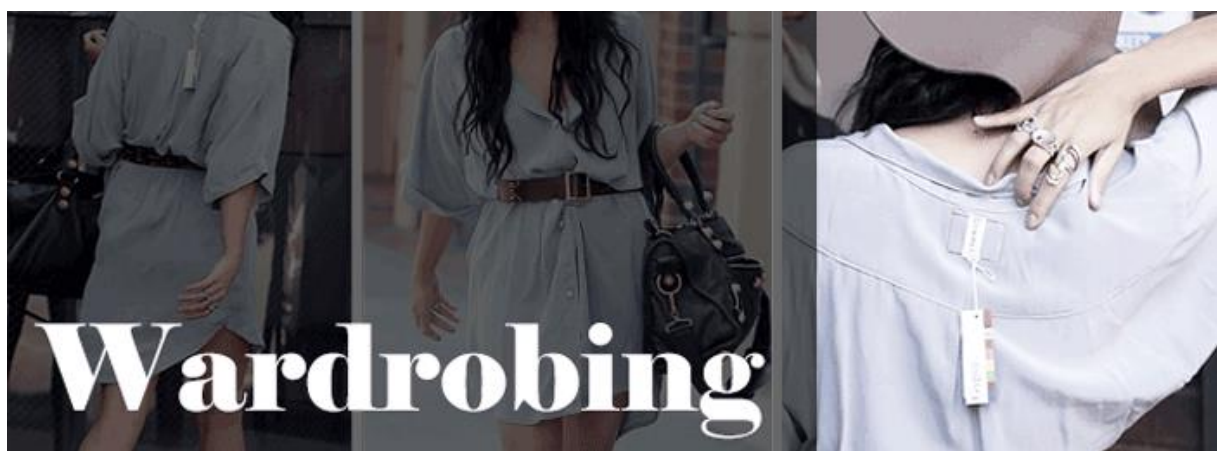
This one seems similar to the unmatched description, but in fact it's quite different. In the case of the misleading description, the problem comes from the contrast between the product's description and the product itself, while in these case it comes from the difference between the product and the customer's expectations. You would be correct to say that the expectations are built from the product details, but not only. The price of the product also affects the expectations: the higher the price, the higher the quality, is what a customer assumes. However, this is not always the case and in order to avoid this, the best piece of advice we can give you is not to overprice your products. If the quality of your products is high, keep your prices high as well, but the opposite should be applied too.

In these cases a store credit or a full refund will be the best applicable refund policy.



Having shipped the wrong item is a common situation as well, but the fault lies entirely on the retailers. In order to avoid such mistakes, make sure that your staff is sorting and shipping the correct items.

In this case, you can offer an exchange for the correct item.



*Wardrobing* is the practice of purchasing an item, using it for a while, and then returning it back to the store for a refund. In general, the most common items that are being *wardrobed* are clothes and accessories, but the same practice may apply to electronics and other tools. The only way you can keep your products from being *wardrobed* is creating a return policy which states the terms and conditions under which you are willing to refund products e.g. with their tags still on, all original packaging (boxes, bags, etc)

should be returned, clean and not damaged in any way items, refund within 10 days, etc. As a store owner, you should keep in mind that there would be customers who would always try to find holes in your refund policies, take advantage and abuse them, so make sure that they are clear, strict and detailed enough.

In the second part of the blogpost, we will share with you some real retail stories for customers willing to return their purchased items for a refund and the reasons behind it. You might find some funny, other absurd and third even outrageous, so keep on reading and be prepared for anything your customers might come up with.

## 7 bizarre stories for customers willing a refund

### The broken dishwasher



Some time after the purchase of a dishwasher, a customer came back to the store asking for a refund, reporting that the *“The dishwasher is quite obviously faulty – when set to wash, water sprays, but the plates don’t spin”*

## The non-waterproof jacket



An angry client came into a store to return a jacket he bought and threatened to escalate his complaint, since although the jacket was labeled to be ‘waterproof’, it didn’t protect his phone put in the pocket of the jacket from being corrupted after a full wash cycle.

## The defective monitor



A helpdesk agent received a complaint from a lady, who claimed that she was able to view her monitor in the right way only when she laid her head on the desk. Once the monitor was rotated by the support team, her issue was solved.

## A grill with a bonus





A guy came into a store to return a grill he bought two days ago. When asked whether he wants to exchange it for another one, he rejected the offer by saying that the grill wasn't working and that he would never buy this product again. When the sales associate started processing the returned grill, he found a grilled steak still laying on the top.

## The borrowed jeans



A lady tried to return a pair of jeans bought a while ago without having neither a receipt nor a tag. The jeans were stained all over the crotch area and when being told that she is not able to get a refund if the jeans are in such a condition, she justified herself by stating that the stains weren't hers and that she hasn't worn them. It was her roommate who borrowed it from her without asking.

## The \$17 000 bill



A customer went into a store and spent \$17 000 in one day. On the next day, the customer returns with a note from a therapist claiming that the customer is a shopping addict and the bought items have to be returned.

## Sand&Salt

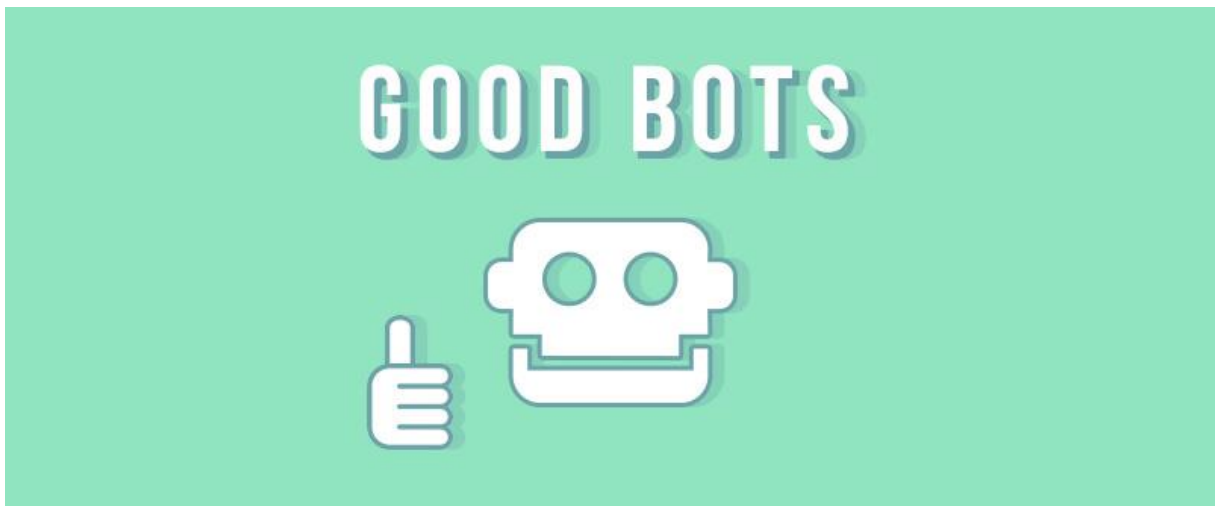


An elderly couple used to buy a Famous Brand's pudding powder, use the pudding powder and replace it with a mixture of sand and salt, pack it nicely and return it back to the store for a refund. They managed to do so in four stores, returning more than 50 packages of pudding before they got caught.

## Final Words

We hope you enjoyed our blogpost! Feel free to share your stories, tips and ideas on how to minimize the returned products as well as how to construct a refund policy, which is beneficial both for clients and for retailers

# The essential guide to good bots, content harvesters and comment spammers

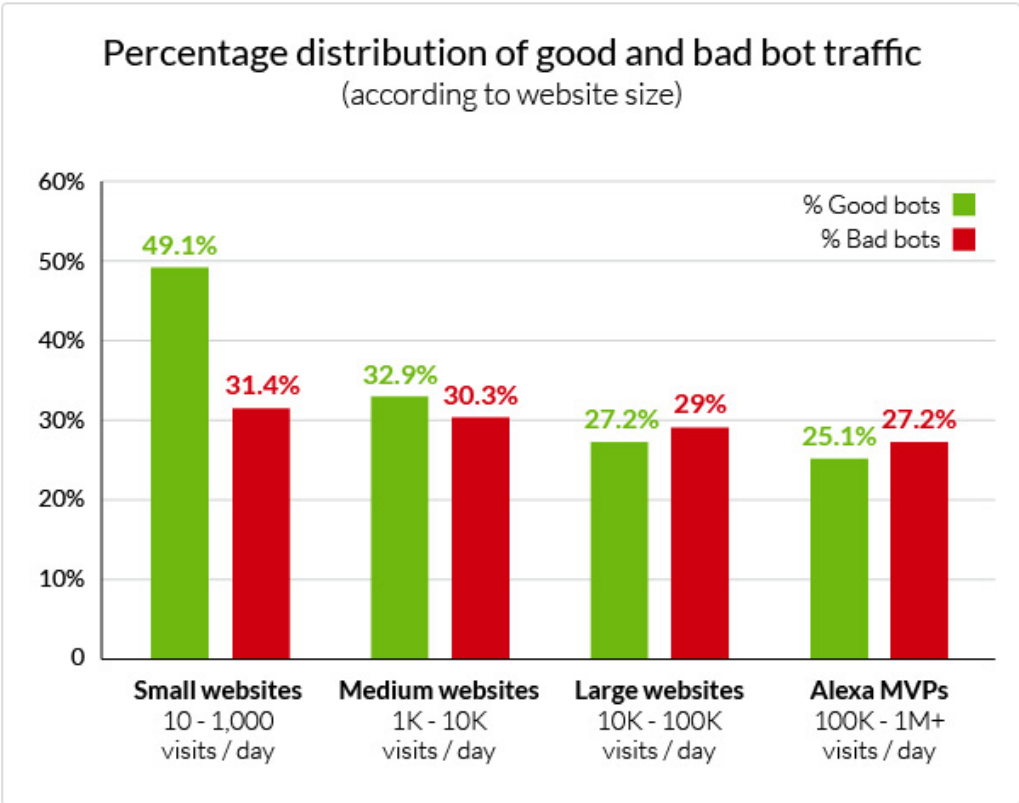


Based on the [report of Incapsula](#) in 2014, 56% of the internet traffic was generated by bots. Bots are considered to be automated software applications performing repetitive work in Internet. These programs usually perform the most mundane and time consuming tasks. In this article we are going to present the two faces of bots usage: benefits of properly using the good bots and the awareness and potential protection from the bad bots: content harvesters and comment spammers.

## Good bots vs Bad bots

If you run an e-commerce site, manage a company website or own a personal website, you have probably heard about the impact of bots in the online world. Bots are crawling and running on internet in 24/7, visiting sites and performing different tasks. One of the most important task performed by bots is search engine indexing.

These bots are called Good bots, known as search engines crawlers and spiders, the most important of which are: Googlebots, Baidu Spider,MSN Bot/Bingbot etc. On the other hand, according to the [Incapsula report](#), nearly 3 out of every 10 visitors of your site are trying to steal information, break the security tools and pretend to be something they are not. These are known as bad bots, the most common of which are content harvesters and comment spammers.



Source: 2014 Bot Traffic Report:  
<https://www.incapsula.com/blog/bot-traffic-report-2014.html#sthash.o7cfB0oe.dpuf>

### Good bots

Good bots can simply be called as programs that search engines launch to get their database indexed with the relevant and viable websites. Even though there are several important Good bots

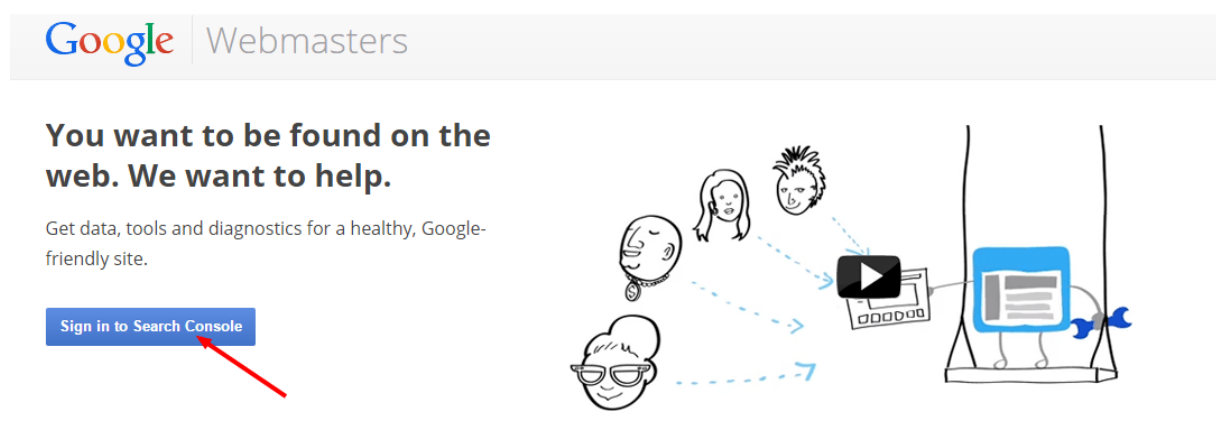
running in internet, one with the most impact in your site performances is the Googlebot. A "Googlebot" is Google's web crawling bot, that crawls the Internet in search of new pages and websites to add to Google's index. In order to get the most advantage of the Googlebots your site needs to be SEO friendly. There are several ways how to make sure that your site it getting the best indexing by Googlebot. You can check the performance of Googlebot running in your site by using the statistics provided by [Google Webmaster Tools](#).

## Verify your site with Google Search Console

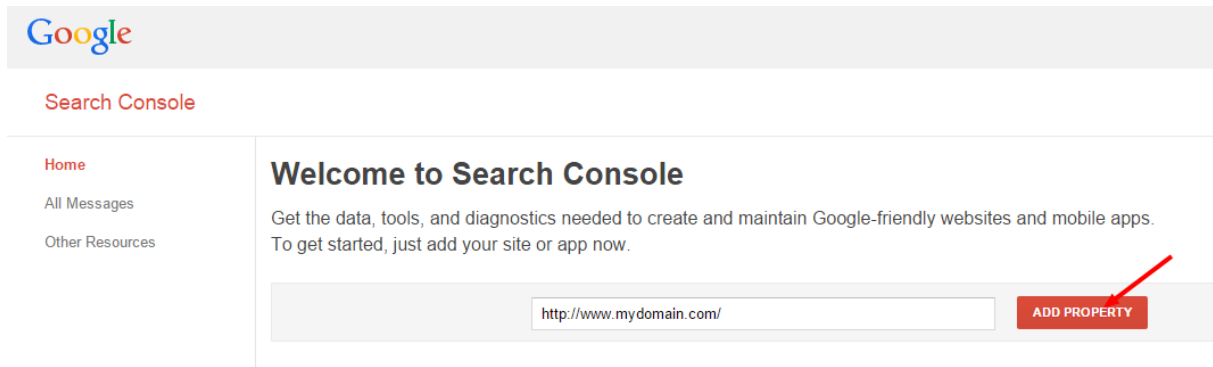
Below you will find a small step-by-step tutorial how to verify your site with Google Webmaster Tools, recently renamed as Google Search Console.

### Step 1: Register for Google Webmaster tools (Google Search Console).

Login to your Google account and access the [Google Webmasters tools](#)

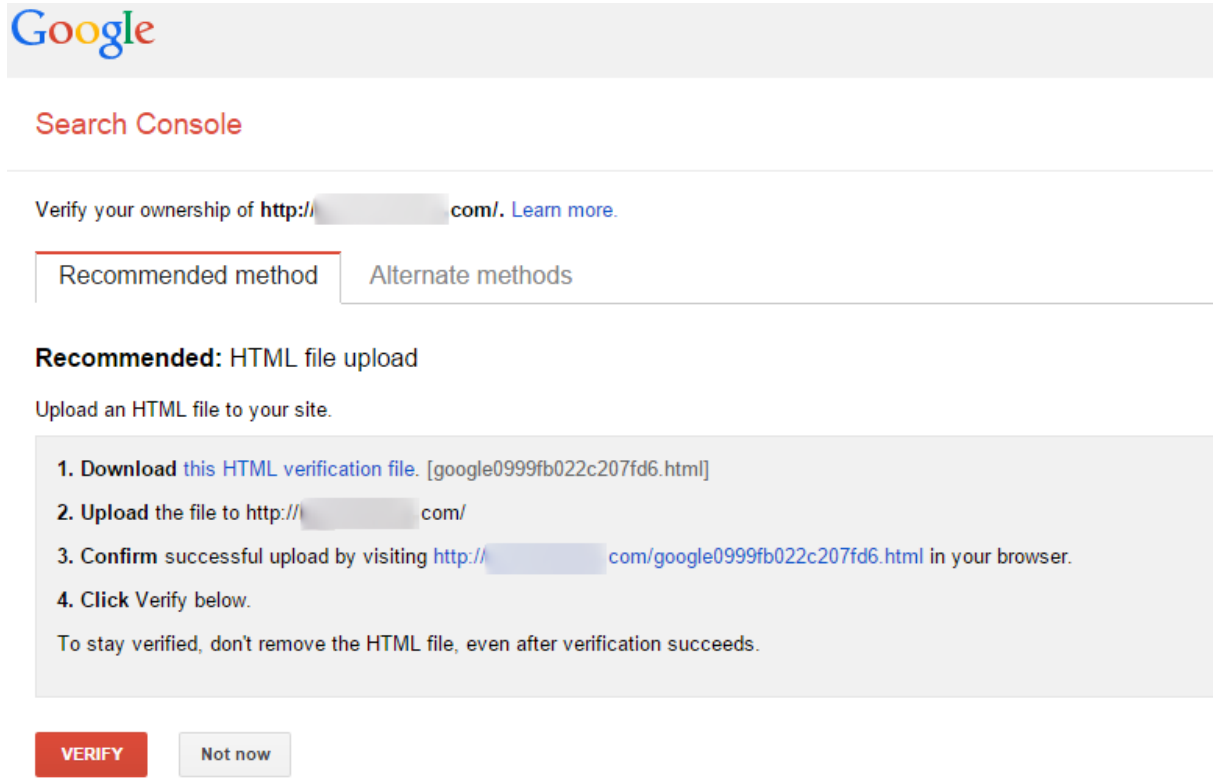


Enter your website domain name in the box and click Add Property.



## Step 2: Verify the website ownership

Perform the instructions stated in the picture below to make sure for the website ownership



If all the steps are passed successfully, when you press the **Verify** button you will see this message:



Congratulations, you have successfully verified your ownership of <http://www.123.com/>.

[Continue](#)

Googlebot will start to review your site and will help Google update search results with your new content. However, your website won't get instantaneously indexed and available through Google Search. The indexing process may take between 48 hours and a few days to complete.

If you own an OpenCart store, you can use [SEO Pack Pro](#) module in order to get the best advantages from the SEO Good bots.

Furthermore we suggest you to follow these articles:

<http://isenselabs.com/posts/boosting-sales-in-opencart-seo-tips-and-social-media-activity-part-1>

## Bad bots

Based on the [Incapsula report](#) , the internet malicious bots consist of four types: Impersonators, Hacking Tools, Scrapers and Spammers. These bots compose  $\frac{1}{3}$  of all the sites' traffic.

Impersonators are the more sophisticated DDoS, ad fraud, and malicious scanning bots, because they try to appear as legitimate users. These bots have in their target all types of websites. Hacking tools on the other hand are focused more on the CMS-based websites (Wordpress, Drupal...). The damage they commit consist in data theft and [site/server hijacking](#). E-commerce sites have a bigger risk to get attacked by scrapers bots. These bots tend to harvest content and perform reverse engineering of pricing and business

model of these sites. Spammers, composing 0.5% of all the websites visits tend to do comment spamming, phishing links and search engine blacklisting. For the rest of this article, we are going to analyse the most frequent bots on e-commerce sites: Scrapers and Spammers.

## Content Harvesters

These bots are considered to be the web scrapers/harvesters, automated programs that extract informations from websites. This is not necessarily a bad activity; for instance price comparison sites rely on the technique. On the other hand, if you are putting valuable content online, these Content Harvesters could pose a real threat to your business. The most common industry these bots attack is e-commerce sites. They use the stolen content to intercept web traffic. Furthermore, they merge different content to make new content so they can avoid duplicate content penalties. Beside stealing your content, these bots can penalize your SEO ranking. Your website could be hit by several penalties for duplicate content and even fail to appear in search engine rankings.

In order to prevent these attacks before they cause damage you need to be more proactive to your site. Preventing is better than curing.

In order to detect if your site has duplicated content you can use online tools to detect plagiarism. One alternative can be [Copyscape](#). What you have to do is enter your site URL and check for duplicated content.



# Copyscape

Search for copies of your page on the web.

Watch the [Video Intro to Copyscape](#) **New!**

PROTECTED BY **COPYSCAPE** DO NOT COPY

Defend your site with a free [plagiarism warning banner!](#)

**Premium** - Check if content is unique and original. **Copysentry** - Monitor the web regularly for plagiarism.

[Share...](#) [Products](#) [Plagiarism](#) [Press](#) [Testimonials](#) [Example](#) [About](#) [Log In](#)

Copyscape © 2015 Indigo Stream Technologies, providers of Giga Alert. All rights reserved. Terms of Use.

If you have identified the scraper bots by their IP, you can block their IP address directly from your .htaccess file. All you have to do is add the following code :

```
RewriteEngine on
RewriteCond %{REMOTE_ADDR} ^69.16.226.12
RewriteRule ^(.*)$ http://newfeedurl.com/feed
```

Where 69.16.226.12 is the banned IP,  
and <http://newfeedurl.com/feed> is the redirecting URL.

For more advanced techniques on how to be protected from harvester bots follow this [article](#).

## Comment Spammers

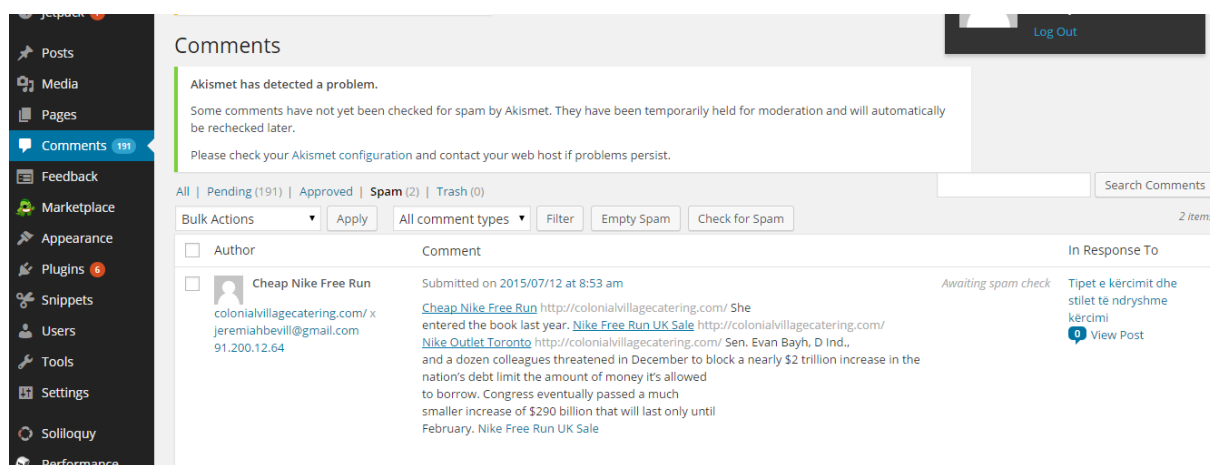
It is likely that in your website forum, blog or comment section you have detected some suspicious comments, usually unrelated with the discussed topic. These unwanted comments are usually spam content created by the comment spammers bots.

*Note: there can also be human comment spammer*

By publishing in your site, spammers have these benefits:

1. They can achieve a slightly higher search engine ranking.
2. Generating traffic and sometimes even making real sales.

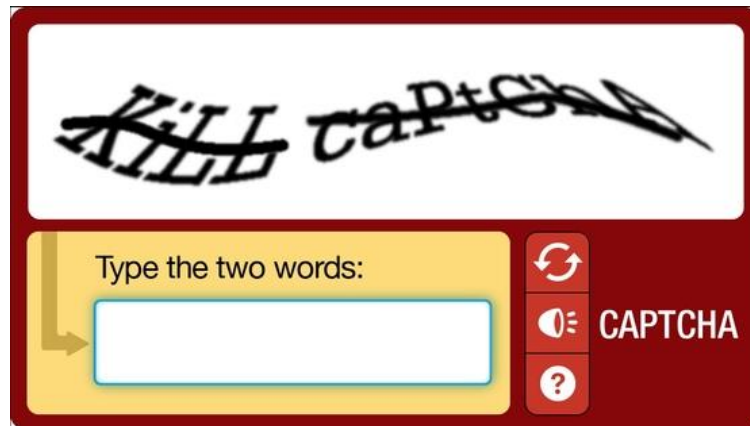
The purpose of spammer is not to degrade your site, what they want is to make more profit. The figure below shows an example of a spam comment in a Wordpress site:



There are several security measures that can protect your site from comment spammers. These techniques try to stop the spam before the comment is posted. Below we will be present two of the most common techniques of protecting from comment spammers.

## Use of CAPTCHA

One of the most popular security measures today is CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) codes part of [Turing Test](#). A CAPTCHA code is usually an image with randomly generated letters and numbers. The content must be entered in the box in order to complete the registration process or to post comments.



By passing this process, it is guaranteed that the commenter is a human not a bot. On the other hand, this technique sometimes appears annoying for a commenter and may discourage the user to post a comment.

## Authentication use

This is also a simple approach, when the site owner requires from the user to provide a username and a password before posting a comment. If the administrator then finds a user spamming the site, he or she can ban that username or their corresponding email address.

In this way the number of potential spammers will diminish since spammers do not want to be identified and spend a lot of time to sign up. Therefore, they would rather attack easy targets. Like in the CAPTCHA case, this method has the disadvantages of discouraging the user to post a comment.

## Social logins authentication

To overcome the problems of the traditional way of authentication, in the recent years a new form of authentications has gain popularity. Based on a [research by Janrain and Blue Research](#) in 2011, 77 percent of the customers prefer social login way of authentication comparing with the traditional one. Social logins as a form of [single sign-on](#), allows the users to login in a website using

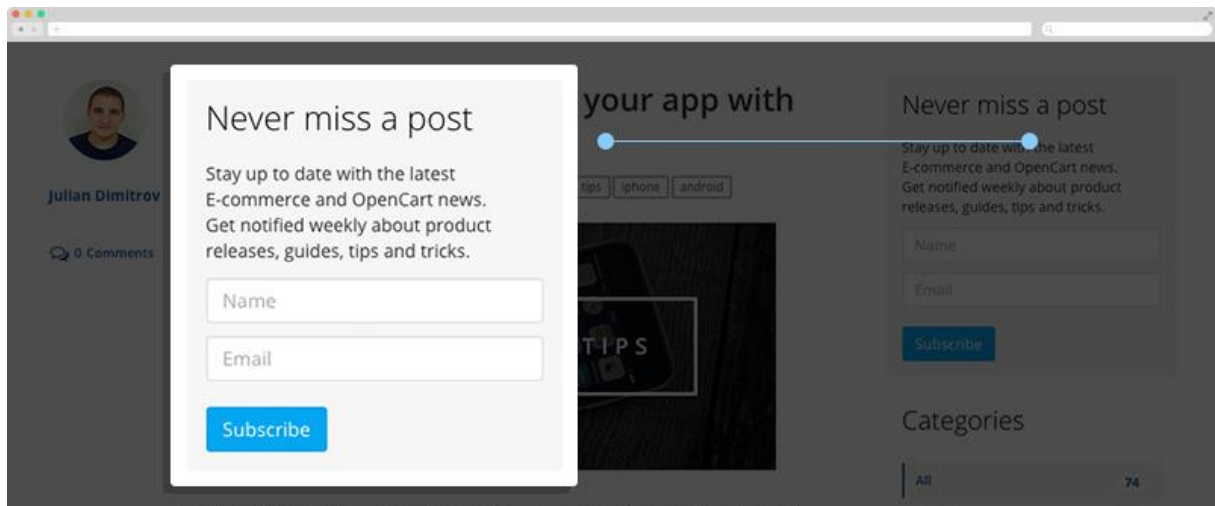
their social networking profiles. Being single sign-on, the user do not need to enter their authentication information multiple times, as the login credentials are remembered in multiple sites. When authenticating with Facebook, the user profile information is automatically retrieved and just a button press can automatically complete the registration process. Based on a [research by Gyga](#), the the majority of social networkers worldwide used their Facebook IDs for social sign-in.

## Conclusions

In order to maintain the success rate of your website you need to be cautious of the human generated bots crawling the internet. To get the quality traffic you need to optimize for the Good bots and try to exclude the Bad ones. Make your site SEO friendly, protect it from content harvesters and take into consideration the common spamming techniques to avoid most spams.

If you are running an OpenCart store, in order to take the most advantages of the clawing bots we recommend [SEO Pack Pro](#) extension. Further, to protect your store from [unwanted bots](#) use [BotBlocker module](#).

# The Latest OpenCart and Ecommerce News Delivered to your Email Inbox



At iSenseLabs we love to learn and we love to teach. We also love being productive, we love exploring and coming up with new ideas. We love helping our customers solve real business problems just as we love providing support for our products. We love tech, gadgets and e-commerce and we love the speed at which things are unwrapping and developing in the industry. We love going into new and unexplored areas. We love creating value and sharing it.

We totally understand that everyone is busy running up their store, blog, business, venture or just has too much things to catch up with. This is why we introduced an easy way for you to make sure you never miss a any information we share in our blogpost. We will email you on Monday and you will get a recap of everything that you might have missed during the past week. Subscription is open and free for anyone (both iSenseLabs existing accounts and users not registered on our website) and takes 10 seconds. If you feel like

you can't catch up with emails you can unsubscribe at any time. Sounds good? Why don't you register right now and get notified on the latest E-commerce and OpenCart news. product releases, guides, tips and tricks, delivered straight to your inbox.