

**User guide**

Prestashop Module

SUMMARY

[Introduction 1](#_Toc399254803)

[Installation 1](#_Toc399254804)

[First use 1](#_Toc399254805)

[Connection 1](#_Toc399254806)

[Customer data binding 1](#_Toc399254807)

[Different types of fields 1](#_Toc399254808)

[Constraints 1](#_Toc399254809)

[MailPerformance forms in Prestashop 1](#_Toc399254810)

[Form embedded in page 1](#_Toc399254811)

[Form on dedicated page 1](#_Toc399254812)

[Prestashop event handling 1](#_Toc399254813)

[Insertion in segment 1](#_Toc399254814)

[Modifiation of target 1](#_Toc399254815)

[Abandoned Cart Erreur ! Signet non défini.](#_Toc399254816)

[Adding features manually 1](#_Toc399254817)

[Modification of the Prestashop customer card and setting in red list 1](#_Toc399254818)

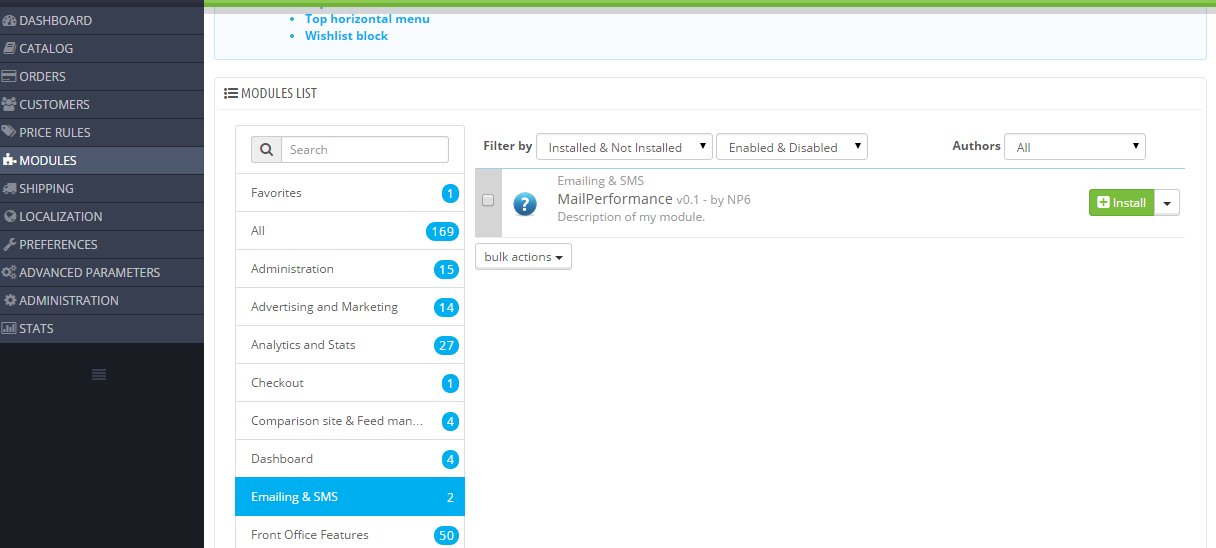
# Introduction

This document is the user guide for the MailPerformance Prestashop module. It explains how to install the module in Prestashop and how to use it.

# Installation

To install the module in PrestaShop download it and follow the steps:

1. If you downloaded the module as a zip file, extract the zip in the modules folder of your PrestaShop website. If you downloaded the module from the PrestaShop Addons Marketplace directly, proceed to the next step.
2. In the PrestaShop back office click on the tab “Modules” (left menu) and select the module category: “Emailing and SMS”. You will find the MailPerformance module in the list. Click “Install”. After the module has been installed, its configuration page will be loaded.



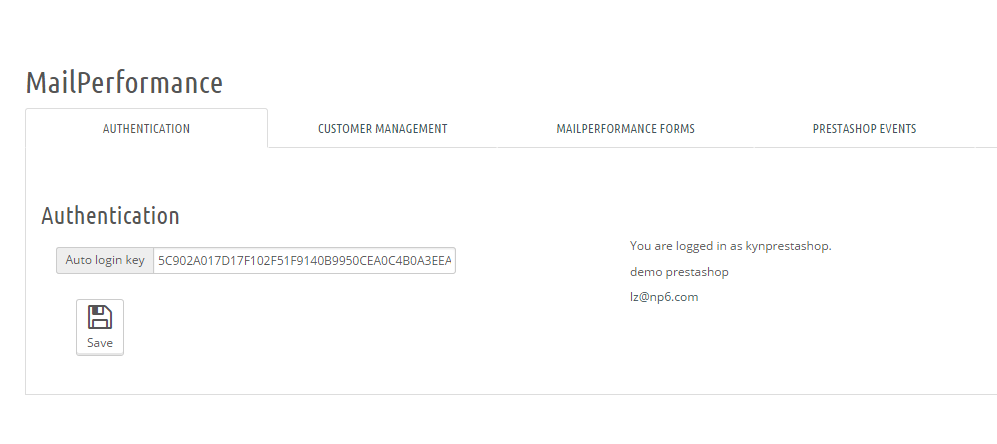
# First use

**Caution** the MailPerformance contact chosen to use the module must have the permissions on form, target, field, values list, segment and contact.

## Connection

The module uses an auto login key to connect and authenticate with the MailPerformance API. The first step in configuring the module is to provide this key. To get your API Key, you have to send an email to [scli@mailperformance.com](mailto:scli@mailperformance.com) with your MailPerformance identifiers.

On the configuration page of the MailPerformance PrestaShop module, copy this key into the field “Auto login key” and click save. If authentication succeeds, you will see your login information displayed and new configuration pages will be available.

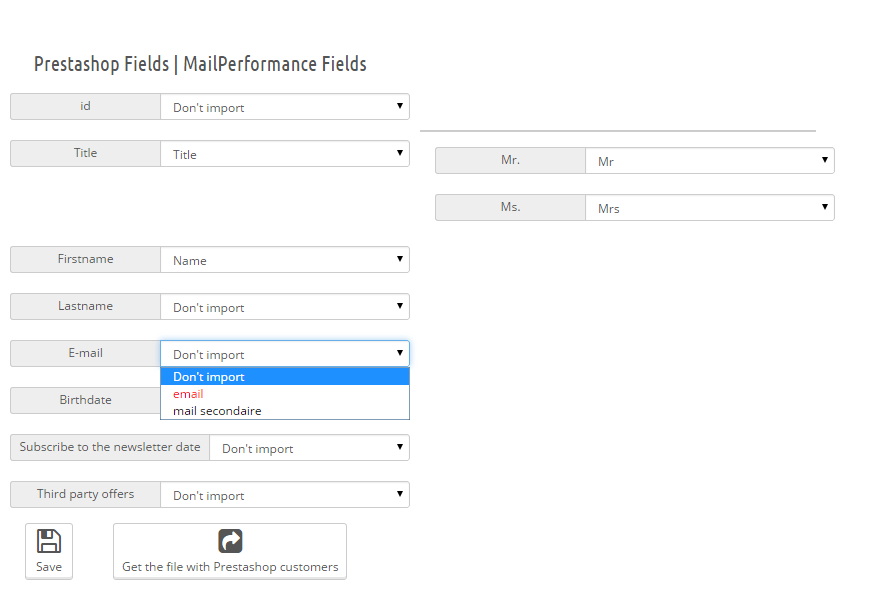


# Customer data binding

The PrestaShop module allows binding MailPerformance fields to PrestaShop customer information. Once a binding is established, the module can add PrestaShop customers to MailPerformance as soon as they are created.

To bind the PrestaShop fields with your MailPerformance fields go to the “Customer Management” tab. Choose the PrestaShop fields you wish to bind with MailPerformance and for each of these fields, select a corresponding MailPerformance field. You do not have to bind all PrestaShop fields, however bindings must exist for MailPerformance mandatory and unicity fields. These obligatory MailPerformance fields are highlighted in red. Saving your bindings will check their validity.

You can also export your existing customers to a CSV file. The file is prepared with the bound fields and a corresponding header. You may use this file to create a manual import in the MailPerformance web interface.



## Different type of fields

You cannot bind any type of PrestaShop field with any type of MailPerformance field. Certain restrictions apply. The module only proposes binding between valid field types.

###### Text fields

You can bind PrestaShop text fields like “first name” and “last name” with MailPerformance Text area or a character string fields.

###### Date fields

You can bind PrestaShop date fields such as “birthdate” and “Subscribe to the newsletter date” with MailPerformance date fields. Should you wish to export existing PrestaShop customers to MailPerformance, you must also choose a date format, which will be used during the creation of a MailPerformance import.

###### List of values fields

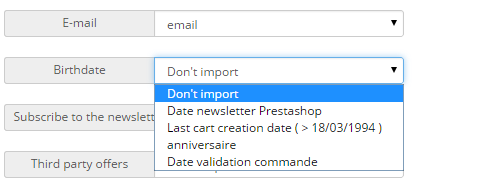
Some PrestaShop fields such as “Third party offers” take a limited number of values (true or false in this case). You can link these fields with a MailPerformance “list of values” or “list of multiple values” field. After choosing the binding, a list of possible PrestaShop field values is displayed. You must bind each of these field values with MailPerformance field values.

###### Email or number field

You can bind PrestaShop fields of type email (“E-mail”) and number (“id”) with their respective MailPerformance field types: email and numeric.

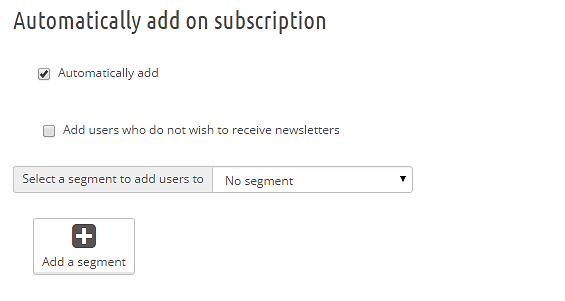
## The constraints

MailPerformance fields can have constraints applied to their values. For example, a date field may be restricted so its value must be more recent than a certain date. PrestaShop does not support this functionality. If you bind a PrestaShop field with a constrained MailPerformance field the module will not verify the constraint on the PrestaShop side. If a PrestaShop customer is created with a field value that breaks the constraint this customer will not be added to MailPerformance as a target.



## Automatic Add

Once a valid binding is in place, you can activate the automatic creation of MailPerformance “Targets” corresponding to PrestaShop customers. Click the checkbox “Automatically add”. When activated, further choices appear. You may choose to create targets for customers who have chosen not to subscribe to your newsletters (in MailPerformance these are red list targets). You can also choose to insert all targets created via PrestaShop into a specific segment. If you have not already prepared such a segment, you may create the segment directly from the module.

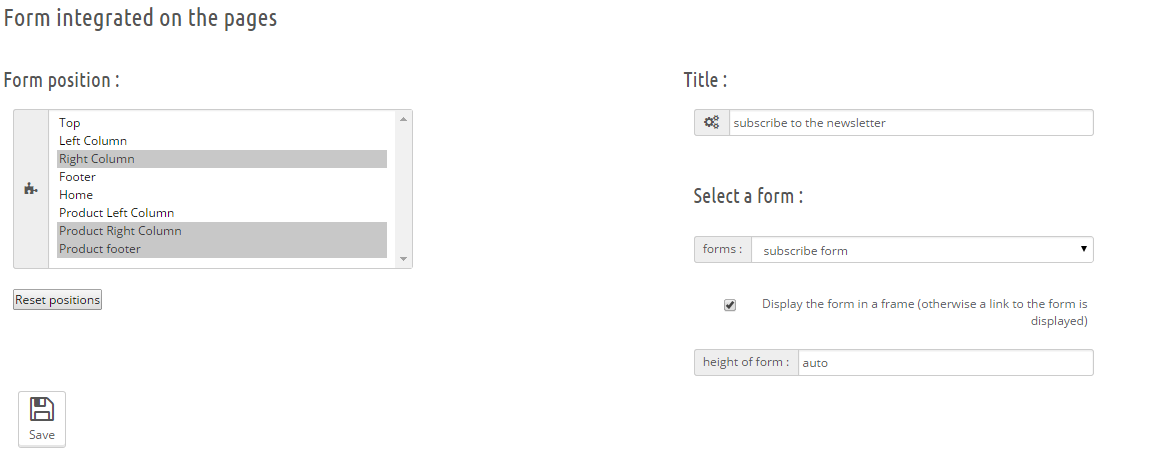


# MailPerformance forms in PrestaShop

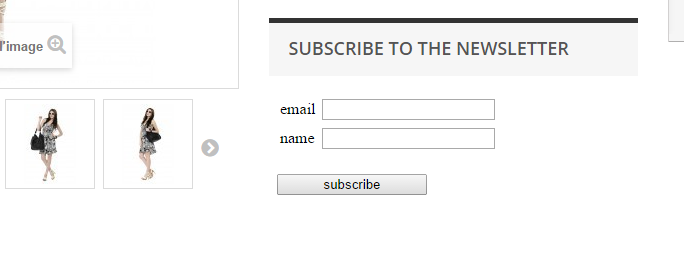
The module allows displaying forms MailPerformance directly in PrestaShop. It offers two ways to display forms: embedded forms and CMS forms. An embedded MailPerformance form is a small form (or a link to a larger form) that is placed at certain locations on your PrestaShop store pages. A CMS form is a PrestaShop CMS page that contains a MailPerformance form that you can use like any other CMS page. You can configure these forms in the MailPerformance forms tab.

## Embedded Form

To display a MailPerformance form embedded in the pages of your store simply choose a form from the list (only standard forms are displayed for other types of forms, [see here](#_Form_on_a)). Choose a title (not required), choose to display a button that will redirect to the form or to imbed the form in a frame. If a button is chosen provide the text label of the button, if a frame is chosen provide the height of the frame. Do not forget to select the locations where the form should be displayed. You may select multiple locations with the Ctrl key.



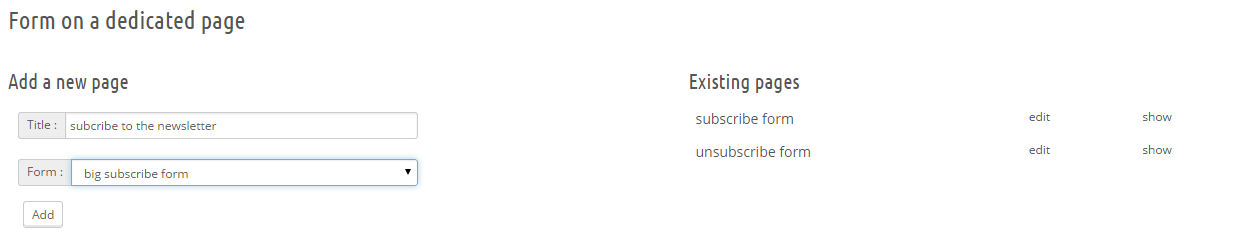
Example of display next to the product:



## CMS Form

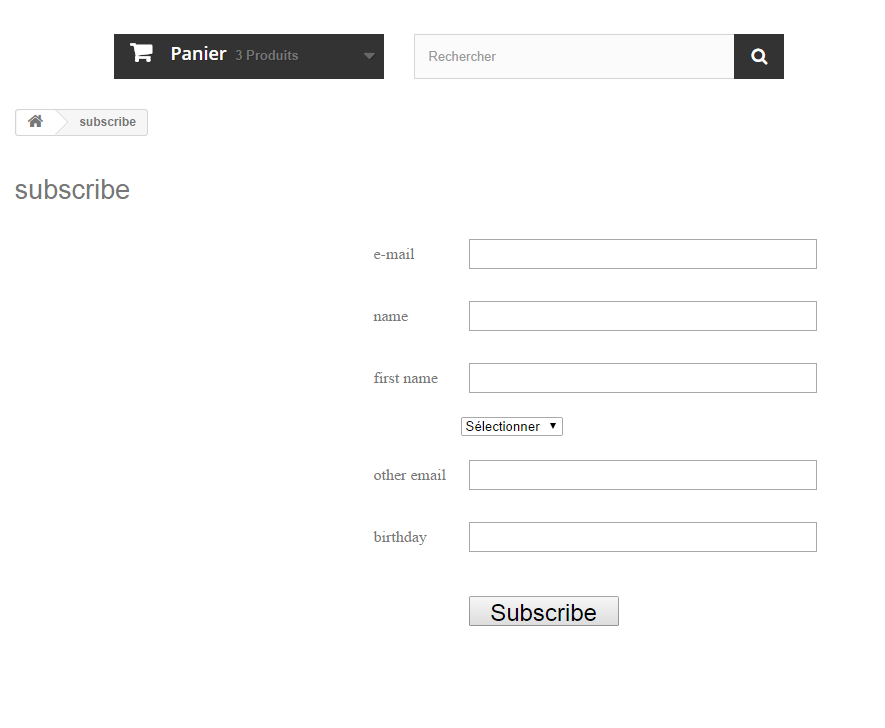
To display other types of forms MailPerformance on your website (unsubscribe, sponsorship) or to display large forms that cannot be embedded, create a dedicated page for your form. This creates a new CMS page on your PrestaShop site. You can use the CMS form like any other PrestaShop CMS page.

To create the CMS form, go to the section “form on a dedicated page”. Enter the title of the page and select a form. Then click on 'Add '.

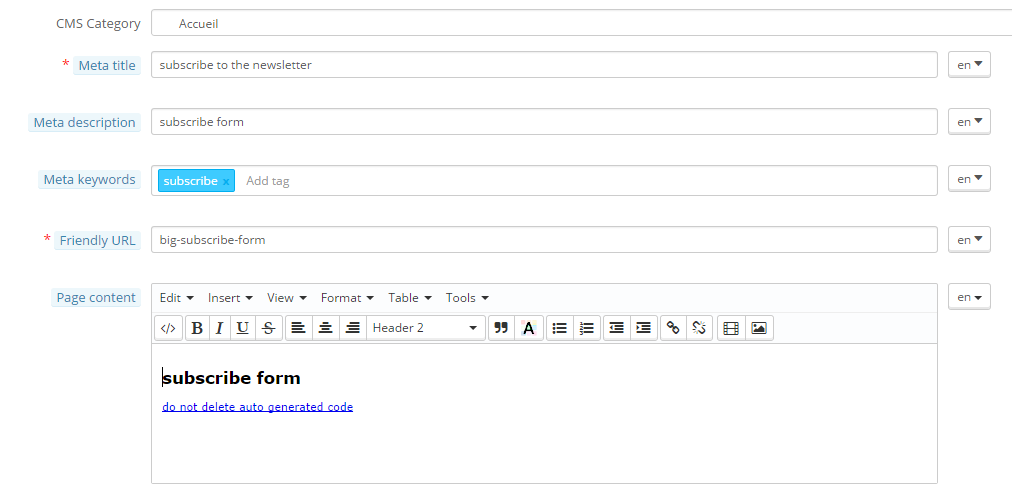


In the picture above on the right, you can see a list of CMS pages that are displayed a MailPerformance form.

CMS Form as displayed by PrestaShop:



CMS modification page (you can modify and add text all around the auto generated code):

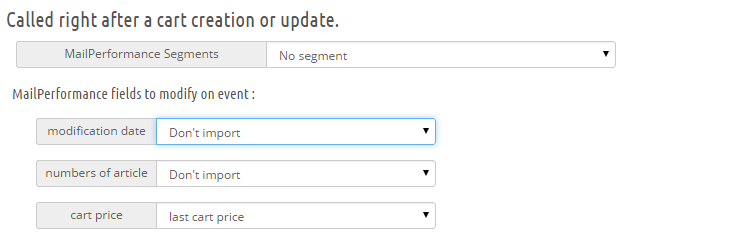


# PrestaShop event handling

The module can handle certain PrestaShop events: cart creation, cart modification, product return, order validation, etc...TODO When these events are triggered by a customer you can insert the corresponding MailPerformance target in a segment or modify some of the target’s field values. You configure the events in the “PrestaShop events” tab.

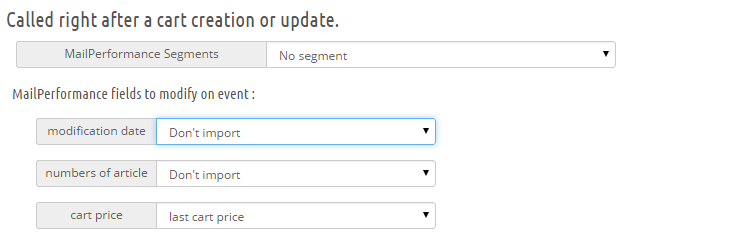
## Insertion in segments

Each PrestaShop event can trigger the insertion of the corresponding MailPerformance target into a specific segment. This can be useful to launch actions or create different scenarios.



## Changing values of the target

Each PrestaShop event proposes different event information that may be inserted into fields of the corresponding MailPerformance target when the event is triggered. Binding event information with a MailPerformance field functions in the same ways as binding customer data.



# Abandoned Cart

The MailPerformance PrestaShop module offers an abandoned cart detection function, permitting e-mail marketers to detect customers who have not fulfilled an order after N-days (where N is typically 1 or 2 days).

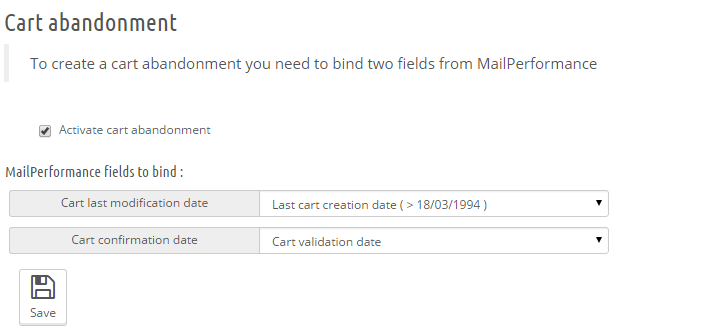
You configure this function in the “Cart abandonment” section.

This is achieved in two steps. The PrestaShop module binds with certain MailPerformance fields and sets them to with specific values. A “cart modification date” field and a “cart validation date” field. When a cart is created or updated, the cart modification date is updated and the cart validation date is set to empty. When a cart is validated, the validation date is set. With these fields, a MailPerformance user can create a dynamic segment corresponding to customers with abandoned carts using the following criteria:

* cart modification date : is before the current date minus 2 days
* cart validation date : is empty

This allows you to schedule an abandoned cart reminder message on the entry in the segment.

To activate the abandoned cart function check the cart abandonment checkbox and select the MailPerformance fields to bind.



# Adding features manually

## Modification of the PrestaShop customer information and setting in red list

By default, PrestaShop does allow a module act when a customer modifies its customer information. For example, by changing email address or unsubscribing from a newsletter. However, you can add this functionality manually by inserting a line in the PrestaShop source code.

Modify the file IdentityController.php in the folder /controllers/front at line 102 (for Prestashop version 1.6.0.4 to 1.6.0.9). Add the line:

Hook::*exec*('actionCustomerAccountUpdate', **array**('customer' => $this->customer));

The file should look as follows:

**if** (!Tools::*getIsset*('optin'))

$this->customer->optin = 0;

**if** (Tools::*getValue*('passwd'))

$this->context->cookie->passwd = $this->customer->passwd;

**if** ($this->customer->update())

{

Hook::*exec*('actionCustomerAccountUpdate', **array**('customer' => $this->customer));

$this->context->cookie->customer\_lastname = $this->customer->lastname;

$this->context->cookie->customer\_firstname = $this->customer->firstname;

$this->context->smarty->assign('confirmation', 1);

}

The field bindings used those configured in the section Customer Data Binding.