# **Perfexcrm info:**

|  |  |
| --- | --- |
| **High Resolution** | [Yes](https://codecanyon.net/attributes/high-resolution/yes) |
| **Compatible Browsers** | [Firefox](https://codecanyon.net/attributes/compatible-browsers/firefox), [Safari](https://codecanyon.net/attributes/compatible-browsers/safari), [Opera](https://codecanyon.net/attributes/compatible-browsers/opera), [Chrome](https://codecanyon.net/attributes/compatible-browsers/chrome), [Edge](https://codecanyon.net/attributes/compatible-browsers/edge) |
| **Files Included** | [JavaScript JS](https://codecanyon.net/attributes/source-files-included/javascript%20js), [HTML](https://codecanyon.net/attributes/source-files-included/html), [CSS](https://codecanyon.net/attributes/source-files-included/css), [PHP](https://codecanyon.net/attributes/source-files-included/php), [SQL](https://codecanyon.net/attributes/source-files-included/sql) |
| **Software Framework** | [CodeIgniter](https://codecanyon.net/attributes/software-framework/codeigniter) |
| **Software Version** | [PHP 8.x](https://codecanyon.net/attributes/compatible-software/php%208.x), [MySQL 8.x](https://codecanyon.net/attributes/compatible-software/mysql%208.x), [MySQL 5.x](https://codecanyon.net/attributes/compatible-software/mysql%205.x) |

#### **System Requirements**

* PHP >= 8.1+
* MySQL 5.1+
* mod\_rewrite Apache
* MySQLi PHP Extension
* PDO PHP Extension
* cURL PHP Extension
* OpenSSL PHP Extension
* MBString PHP Extension
* iconv PHP Extension
* IMAP PHP Extension
* GD PHP Extension
* Zip PHP Extension
* allow\_url\_fopen enabled

In most hosting accounts these extensions are enabled by default. But you should consult with your hosting provider.

# **Introduction to Perfex CRM modules**

/[Modules](https://help.perfexcrm.com/category/modules/) /Introduction to Perfex CRM modules

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

The modules documentation is valid starting from version **2.3.2**

Perfex CRM [version 2.3.0](https://help.perfexcrm.com/version-2-3-0/) comes with a built-in feature to create modules and integrate additional features into the core functionalities.  
  
If you are eager to create modules for Perfex CRM to sell them on CodeCanyon or create your own features that you need for your requirements you are in the right place.

Modules are packages of code that extend the core functionality of Perfex CRM , by making your own module you are extending Perfex CRM or integrating brand new features.

The modules can consist of action [hooks](https://help.perfexcrm.com/module-basics/#adding-hooks), filters, [controllers](https://help.perfexcrm.com/module-controllers/), models, custom php classes/libraries, [language files,](https://help.perfexcrm.com/module-language/) [views](https://help.perfexcrm.com/module-views/), this code base is already used in Perfex CRM.

## **What should you know before start creating modules?**

### **Maintaining Modules**

**It’s your responsibility** to maintain the modules you will be creating when there will be Perfex CRM major changes and make them compatible with the latest version.

Perfex CRM is still **a new modular application**, in each update, we will try to enhance the code and make it more modular and for this reason, there will be breaking changes which you will need to adapt your modules to those breaking changes.

### **Modules Security**

Perfex CRM is built with the Codeigniter PHP framework which we follow security practices in order to make Perfex CRM reliable.

When creating new modules, **you must ensure** that your modules are [secure](https://help.perfexcrm.com/module-security/) and use already existing functions and methods that exist in the Codeigniter PHP framework.

### **Selling modules on CodeCanyon**

If you want to create modules and sell them as an [author](https://author.envato.com/join)on [CodeCanyon](https://1.envato.market/Y0GEe), you are free to do so, keep in mind that because of our Exclusivity agreement, you can **ONLY**sell the modules on CodeCanyon.

Selling Perfex CRM modules CodeCanyon means that you are responsible for setting the appropriate price of the module, the way you will present the module, offering support, ensuring that your module is compatible with latest version changes and your marketing strategy.

You can’t present us as a partner to you as an individual or company, publishing a module from your Envato account, means that you own all the copyright to the module and the code.

Under our own discretion, we may decide to market your module or recommend it to our customers.

### **Support for creating modules**

**We won’t offer you support for creating modules**, if you are certain that something does not work right and it’s a bug, you can open a support ticket at [https://my.perfexcrm.com](https://my.perfexcrm.com/) and provide more information, we will be happy to take a look.

However, you should not expect on How To’s to create a specific feature, modules development is not included in support, try to figure on your own, or if you need any specific functions, you should explore the code.

# **Module Basics**

/[Modules](https://help.perfexcrm.com/category/modules/) /Module Basics

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

The modules documentation is valid starting from version **2.3.2**

Perfex CRM modules use the Codeigniter PHP framework code base like controllers, models, language files, libraries, views etc… if you are already familiar with the Codeigniter framework, you will be able to easily get started, if not we recommend you to visit their official [website](https://codeigniter.com/)and [documentation](https://codeigniter.com/docs).

Additionally, we added a bunch of helper functions, features, action hooks and filters that will make the process of creating and integrating modules easier.

### **Modules location & Name**

All modules should be added in the **modules**folder in your root directory where Perfex CRM is installed and **each module must have unique folder name and init file with the same name as your module folder name**.

### **Creating your first module**

Before start to develop a module, make sure that you set [development mode](https://help.perfexcrm.com/faq/how-to-set-debug-mode/) in order to see any errors and functions/hooks deprecation warnings.

To get started and create your first module, follow the steps below.

* Navigate to Perfex CRM installation and open the **modules**directory
* Create a new directory and name it after your module (e.q. **sample\_module**)
* Create a new PHP file **with the same name as the module directory**, in this case, **sample\_module.php,**this file will act as “**init file”** for this module.

In this case, the module is already created and registered but it does not do anything yet nor have any name or descriptions added, we will add the module metadata like name, description, URL, author via a PHP block comment which will act as module headers.

The module headers should be added only in the init module file.

Now open the sample\_module.php file we created before and add the following comment at the top of the file:

<?php

/\*\*

\* Ensures that the module init file can't be accessed directly, only within the application.

\*/

defined('BASEPATH') or exit('No direct script access allowed');

/\*

Module Name: Sample Perfex CRM Module

Description: Sample module description.

Version: 2.3.0

Requires at least: 2.3.\*

\*/

After you save the file, you can navigate in your Perfex CRM admin area to **Setup->Modules** and you will be able to see your module listed.

The comment will act as a file header for the module, you should check the [file headers guide](https://help.perfexcrm.com/module-file-headers/) if you want to get more familiar.

Make sure that you always add the **Module Name** header



It’s very important to **make sure that your module folder name and the .php file name are the same**, otherwise, your module won’t be shown in the modules list

### **Basic Hooks**

You will need to implement few basic hooks when creating new module, like [register\_activation\_hook(),](https://help.perfexcrm.com/common-module-functions/#register_activation_hook) [register\_deactivation\_hook()](https://help.perfexcrm.com/common-module-functions/#register_deactivation_hook) and [register\_uninstall\_hook()](https://help.perfexcrm.com/common-module-functions/#register_uninstall_hook)

If you are familiar with WordPress, you will probably know the work that these hooks do.

### **Adding Hooks**

You can add hooks in your module .php files with:

hooks()->add\_action($tag, $function\_to\_add, $priority = 10, $accepted\_args = 1);

hooks()->add\_filter($tag, $function\_to\_add, $priority = 10, $accepted\_args = 1);

hooks()->do\_action($tag, $arg = '');

hooks()->apply\_filters($tag, $value, $additionalParams);

You can add your own filters and actions so you can use them in your module implementation also, adding actions and filters will help other modules to interact with your module and extend the module functionalities.

### **Prefix Custom Functions**

You should **always prefix any custom functions** and classes to prevent any conflicts with Perfex CRM default functions or classes or with any other modules.

### **Using Codeigniter Instance in module files**

Probably you will want to use the Codeigniter framework instance in your module init file or any other modules files that are not extending the framework base classes.

**$this**, only works within your controllers, your models, or your views.

You can achieve this anywhere in your files with the following code:

First, assign the CodeIgniter object to a variable:

$CI = &get\_instance();

Once you’ve assigned the object to a variable, you’ll use that variable *instead* of**$this**

$CI =&get\_instance();

$CI->load->helper('module\_name/helper\_name');

$CI->load->library('module\_name/library\_name');

### **Database Prefix**

From version 2.3.0 Perfex CRM have option to define custom database tables prefix.

The default table prefix in Perfex CRM is **tbl**

You should always use our custom function **db\_prefix()** to get the database prefix when querying the database, this will ensure that in case the user changed the database prefix, your module will still work with the new user prefix.

Here is an example from the **Goals Tracking** module **install.php** file.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

if (!$CI->db->table\_exists(db\_prefix() . 'goals')) {

$CI->db->query('CREATE TABLE `' . db\_prefix() . "goals` (

`id` int(11) NOT NULL,

`subject` varchar(191) NOT NULL,

`description` text NOT NULL,

`start\_date` date NOT NULL,

`end\_date` date NOT NULL,

`goal\_type` int(11) NOT NULL,

`contract\_type` int(11) NOT NULL DEFAULT '0',

`achievement` int(11) NOT NULL,

`notify\_when\_fail` tinyint(1) NOT NULL DEFAULT '1',

`notify\_when\_achieve` tinyint(1) NOT NULL DEFAULT '1',

`notified` int(11) NOT NULL DEFAULT '0',

`staff\_id` int(11) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=" . $CI->db->char\_set . ';');

### **Create Module Options**

Perfex CRM has table in database **options**for storing various settings for internal usage and settings that are used in features. We have developed custom PHP functions that will perform queries to fetch options from database.

Option names are strings, and they must be unique so that they do not conflict with either Perfex CRM or other Modules.

add\_option($name, $value, $autoload)

**$name**Required (string). The name of the option to be added, make sure it’s unique and prefixed with E.q. your module name.  
  
**$value**The option value (string)

**$autoload**(integer) 1 or 0  
Whether this option should be autoloaded with all other options, if you are using the option too much time in the view, the best is to autoload it to prevent multiple queries in order to get the option. Defaults to 1

Keep in mind that **add\_option**function won’t create the option if the option name already exists in the options table.

get\_option($option\_name);

Retrieve an option from database, **$option\_name** (string) your option name.

update\_option($option\_name, $new\_value);

The **update\_option**function will update the option value, since version 2.3.3, when you call **update\_option if the option not exists, Perfex CRM will create this option.**

# **Module File Headers**

/[Modules](https://help.perfexcrm.com/category/modules/) /Module File Headers

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

Each module in Perfex CRM consist of init file which contains the general module configuration and includes headers containing meta-information regarding the module.

### **Module init file headers example**

The follow example is taken from the default Perfex CRM module Menu Setup that comes with the installation.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

/\*

Module Name: Menu Setup

Description: Default module to apply changes to the menus

Version: 2.3.0

Requires at least: 2.3.\*

\*/

### **Available Headers**

You can find below the list of available headers that can be added to module init file.

* **Module Name** – **required**
* **Module URI** – module URL
* **Version** – the version of the module
* **Description** – the description of the module, you can explain what does the module do.
* **Author**– module author name
* **Author URI**– module author URL
* **Requires at least** – what version of Perfex CRM the module requires at least

# **Common Module Functions**

/[Modules](https://help.perfexcrm.com/category/modules/) /Common Module Functions

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

#### **register\_activation\_hook**

/\*\*

\* Register module activation hook

\* @param string $module module system name

\* @param mixed $function function for the hook

\* @return mixed

\*/

register\_activation\_hook($module, $function)

#### **register\_deactivation\_hook**

/\*\*

\* Register module deactivation hook

\* @param string $module module system name

\* @param mixed $function function for the hook

\* @return mixed

\*/

register\_deactivation\_hook($module, $function)

#### **register\_uninstall\_hook**

/\*\*

\* Register module uninstall hook

\* @param string $module module system name

\* @param mixed $function function for the hook

\* @return mixed

\*/

register\_uninstall\_hook($module, $function)

#### **register\_cron\_task**

/\*\*

\* Register module cron task, the cron task is executed after the core cron tasks are finished

\* @param mixed $function function/class parameter for the hook

\* @return null

\*/

register\_cron\_task($function)

#### **register\_payment\_gateway**

/\*\*

\* Inject custom payment gateway into the payment gateways array

\* @param string $idpayment gateway id, should equal like the libraries/classname e.q. gateways/New\_gateway

\* @param string $module module name to load the gateway if not already loaded

\*/

register\_payment\_gateway($id, $module)

#### **register\_language\_files**

/\*\*

\* Register module language files to support custom\_lang.php file

\* @param string $module module system name

\* @param array $languages array of language file names without the \_lang.php

\* @return null

\*/

register\_language\_files($module, $languages)

#### **module\_dir\_url**

/\*\*

\* Module URL

\* e.q. https://crm-installation.com/module\_name/

\* @param string $module module system name

\* @param string $segment additional string to append to the URL

\* @return string

\*/

module\_dir\_url($module, $segment = '')

#### **module\_dir\_path**

/\*\*

\* Module directory absolute path

\* @param string $module module system name

\* @param string $concat append additional string to the path

\* @return string

\*/

module\_dir\_path($module, $concat = '')

#### **module\_libs\_path**

/\*\*

\* Module libraries path

\* e.q. modules/module\_name/libraries

\* @param string $module module name

\* @param string $concat append additional string to the path

\* @return string

\*/

module\_libs\_path($module, $concat = '')

# **Common Module Functions**

/[Modules](https://help.perfexcrm.com/category/modules/) /Common Module Functions

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

#### **register\_activation\_hook**

/\*\*

\* Register module activation hook

\* @param string $module module system name

\* @param mixed $function function for the hook

\* @return mixed

\*/

register\_activation\_hook($module, $function)

#### **register\_deactivation\_hook**

/\*\*

\* Register module deactivation hook

\* @param string $module module system name

\* @param mixed $function function for the hook

\* @return mixed

\*/

register\_deactivation\_hook($module, $function)

#### **register\_uninstall\_hook**

/\*\*

\* Register module uninstall hook

\* @param string $module module system name

\* @param mixed $function function for the hook

\* @return mixed

\*/

register\_uninstall\_hook($module, $function)

#### **register\_cron\_task**

/\*\*

\* Register module cron task, the cron task is executed after the core cron tasks are finished

\* @param mixed $function function/class parameter for the hook

\* @return null

\*/

register\_cron\_task($function)

#### **register\_payment\_gateway**

/\*\*

\* Inject custom payment gateway into the payment gateways array

\* @param string $idpayment gateway id, should equal like the libraries/classname e.q. gateways/New\_gateway

\* @param string $module module name to load the gateway if not already loaded

\*/

register\_payment\_gateway($id, $module)

#### **register\_language\_files**

/\*\*

\* Register module language files to support custom\_lang.php file

\* @param string $module module system name

\* @param array $languages array of language file names without the \_lang.php

\* @return null

\*/

register\_language\_files($module, $languages)

#### **module\_dir\_url**

/\*\*

\* Module URL

\* e.q. https://crm-installation.com/module\_name/

\* @param string $module module system name

\* @param string $segment additional string to append to the URL

\* @return string

\*/

module\_dir\_url($module, $segment = '')

#### **module\_dir\_path**

/\*\*

\* Module directory absolute path

\* @param string $module module system name

\* @param string $concat append additional string to the path

\* @return string

\*/

module\_dir\_path($module, $concat = '')

#### **module\_libs\_path**

/\*\*

\* Module libraries path

\* e.q. modules/module\_name/libraries

\* @param string $module module name

\* @param string $concat append additional string to the path

\* @return string

\*/

module\_libs\_path($module, $concat = '')

# **Module File Headers**

/[Modules](https://help.perfexcrm.com/category/modules/) /Module File Headers

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

Each module in Perfex CRM consist of init file which contains the general module configuration and includes headers containing meta-information regarding the module.

### **Module init file headers example**

The follow example is taken from the default Perfex CRM module Menu Setup that comes with the installation.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

/\*

Module Name: Menu Setup

Description: Default module to apply changes to the menus

Version: 2.3.0

Requires at least: 2.3.\*

\*/

### **Available Headers**

You can find below the list of available headers that can be added to module init file.

* **Module Name** – **required**
* **Module URI** – module URL
* **Version** – the version of the module
* **Description** – the description of the module, you can explain what does the module do.
* **Author**– module author name
* **Author URI**– module author URL
* **Requires at least** – what version of Perfex CRM the module requires at least

# **Create Menu Items**

/[Modules](https://help.perfexcrm.com/category/modules/) /Create Menu Items

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

If you are creating your custom modules, probably you will want to create menu items that will be shown on admin area sidebar or clients area navigation.

With Perfex CRM you can easily achieve this with few lines of code.

**The code samples below, should be places in the module init file.**

## **Admin Area**

hooks()->add\_action('admin\_init', 'my\_module\_init\_menu\_items');

function my\_module\_init\_menu\_items(){

$CI = &get\_instance();

$CI->app\_menu->add\_sidebar\_menu\_item('custom-menu-unique-id', [

'name' => 'Custom Menu Item', // The name if the item

'href' => 'https://perfexcrm.com/', // URL of the item

'position' => 10, // The menu position, see below for default positions.

'icon' => 'fa fa-question-circle', // Font awesome icon

]);

}

### **Item With SubMenu Items**

hooks()->add\_action('admin\_init', 'my\_module\_menu\_item\_collapsible');

function my\_module\_menu\_item\_collapsible()

{

$CI = &get\_instance();

$CI->app\_menu->add\_sidebar\_menu\_item('custom-menu-unique-id', [

'name' => 'Parent Item', // The name if the item

'collapse' => true, // Indicates that this item will have submitems

'position' => 10, // The menu position

'icon' => 'fa fa-question-circle', // Font awesome icon

]);

// The first paremeter is the parent menu ID/Slug

$CI->app\_menu->add\_sidebar\_children\_item('custom-menu-unique-id', [

'slug' => 'child-to-custom-menu-item', // Required ID/slug UNIQUE for the child menu

'name' => 'Sub Menu', // The name if the item

'href' => 'https://perfexcrm.com/', // URL of the item

'position' => 5, // The menu position

'icon' => 'fa fa-exclamation', // Font awesome icon

]);

}

Make sure to replace the **my\_module** functions prefix with your own **unique** function prefix.

### **Default Admin Menu Items Positions**

The **default menu items have different positions, so you can hook your new items in the middle**, find below the default position, based on where you want to add your custom item, you can adjust the position attribute.

* Dashboard – 1
* Customers – 5
* Sales – 10
* Subscriptions – 15
* Expenses – 20
* Contracts – 25
* Projects – 30
* Tasks – 35
* Tickets – 40
* Leads – 45
* Knowledge Base – 50
* Utilities – 55
* Reports – 60

## **Clients Area**

hooks()->add\_action('clients\_init', 'my\_module\_clients\_area\_menu\_items');

function my\_module\_clients\_area\_menu\_items()

{

// Item for all clients

add\_theme\_menu\_item('unique-item-id', [

'name' => 'Custom Clients Area',

'href' => site\_url('my\_module/acme'),

'position' => 10,

]);

// Show menu item only if client is logged in

if (is\_client\_logged\_in()) {

add\_theme\_menu\_item('unique-logged-in-item-id', [

'name' => 'Only Logged In',

'href' => site\_url('my\_module/only\_logged\_in'),

'position' => 15,

]);

}

}

### **Default Clients Area Menu Items Positions**

* Knowledge Base 5
* Register – 99
* Login – 100
* Projects – 10
* Invoices – 15
* Contracts – 20
* Estimates – 25
* Proposals – 30
* Subscriptions – 40
* Support – 45

# **Common Module Functions**

/[Modules](https://help.perfexcrm.com/category/modules/) /Common Module Functions

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

#### **register\_activation\_hook**

/\*\*

\* Register module activation hook

\* @param string $module module system name

\* @param mixed $function function for the hook

\* @return mixed

\*/

register\_activation\_hook($module, $function)

#### **register\_deactivation\_hook**

/\*\*

\* Register module deactivation hook

\* @param string $module module system name

\* @param mixed $function function for the hook

\* @return mixed

\*/

register\_deactivation\_hook($module, $function)

#### **register\_uninstall\_hook**

/\*\*

\* Register module uninstall hook

\* @param string $module module system name

\* @param mixed $function function for the hook

\* @return mixed

\*/

register\_uninstall\_hook($module, $function)

#### **register\_cron\_task**

/\*\*

\* Register module cron task, the cron task is executed after the core cron tasks are finished

\* @param mixed $function function/class parameter for the hook

\* @return null

\*/

register\_cron\_task($function)

#### **register\_payment\_gateway**

/\*\*

\* Inject custom payment gateway into the payment gateways array

\* @param string $idpayment gateway id, should equal like the libraries/classname e.q. gateways/New\_gateway

\* @param string $module module name to load the gateway if not already loaded

\*/

register\_payment\_gateway($id, $module)

#### **register\_language\_files**

/\*\*

\* Register module language files to support custom\_lang.php file

\* @param string $module module system name

\* @param array $languages array of language file names without the \_lang.php

\* @return null

\*/

register\_language\_files($module, $languages)

#### **module\_dir\_url**

/\*\*

\* Module URL

\* e.q. https://crm-installation.com/module\_name/

\* @param string $module module system name

\* @param string $segment additional string to append to the URL

\* @return string

\*/

module\_dir\_url($module, $segment = '')

#### **module\_dir\_path**

/\*\*

\* Module directory absolute path

\* @param string $module module system name

\* @param string $concat append additional string to the path

\* @return string

\*/

module\_dir\_path($module, $concat = '')

#### **module\_libs\_path**

/\*\*

\* Module libraries path

\* e.q. modules/module\_name/libraries

\* @param string $module module name

\* @param string $concat append additional string to the path

\* @return string

\*/

module\_libs\_path($module, $concat = '')

# **Module Security**

/[Modules](https://help.perfexcrm.com/category/modules/) /Module Security

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

So, you created your module and works fine, but is it secure? You must ensure that your module is secure and is not vulnerable to any SQL Injections are directory traversing.

You can find below best practices to ensure that your module will be secure. Feel free to apply your own best practices for your module security.

### **Gather user data from requests**

When a user is filling forms, the data is sent e.q. via a POST request to the [controller](https://help.perfexcrm.com/module-controllers/) after that you gather this data and insert into the database.

To ensure that this data is escaped, you should gather the data with the built-in CodeIgniter framework input class.

// Get data from POST request

$data = $this->input->post();

$client\_id = $this->input->post('client\_id');

// Get data from GET request

$data = $this->input->get();

$client\_id = $this->input->get('client\_id');

### **Do Not Allow Direct Access the Module Files**

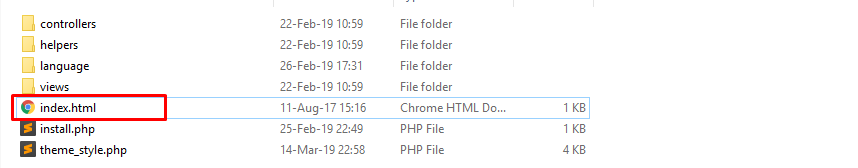
For each .php file you created for your module, you must add the code below at the top of the file to prevent the file to be accessed directly.

defined('BASEPATH') or exit('No direct script access allowed');

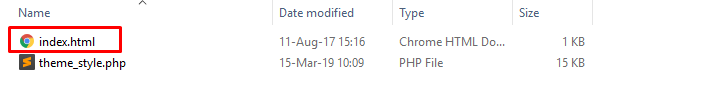
### **Include empty index.html files**

Always add **empty** **index.html**file **in each folder you will create**in your module directory including your module root directory.

E.q. in modules/[your-module]/



E.q. in modules[your-module]/views



# **Module as Payment Gateway**

/[Modules](https://help.perfexcrm.com/category/modules/) /Module as Payment Gateway

* March 24, 2019
* [Modules](https://help.perfexcrm.com/category/modules/)

This feature is available from version 2.3.4

Before the start, we assume that you are already familiar with the [modules basics](https://help.perfexcrm.com/module-basics/), you created a blank module with all the required [headers](https://help.perfexcrm.com/module-file-headers/), and **the module is activated** and shown in the modules list.

* In **modules/[module\_name]/** create folder named **libraries**
* In modules/[module\_name]/**libraries**create a class name e.q. **Example\_gateway.php**, the **filename must end with \_gateway.php**
* See below [**Sample implementation**](https://help.perfexcrm.com/module-as-payment-gateway/#sample-implementation)
* In your module init file, register the payment gateway with the following code:

register\_payment\_gateway('example\_gateway', '[module\_name]');

Replace **[module\_name]**with your actual module system name and **example\_gateway**with your module class name (in lowercase).

If you configured everything as it should, navigate to **Setup->Settings->Payment Gateways**, you will be able to see your **Example**gateway listed there with the options you specified.

Additionally, depends on the gateway API, you can create [a controller](https://help.perfexcrm.com/module-controllers/)to redirect from the **process\_payment**gateway class method.

### **Sample Implementation**

Because Perfex CRM uses Codeigniter framework to integrate the payment gateway you will need to create 1 gateway library (gateway config and process method) and 1 gateway controller (for HTTP requests, show form etc..), you can also take a look at the other gateways files in order to get the idea.

In the invoice HTML area when a customer clicks on the button **PAY NOW**, **we call 1 method from the gateway library** which will process everything additional that is required for this gateway eq redirect to gateway website and pass parameters or redirect to [controller](https://help.perfexcrm.com/module-controllers/)and show form etc…

We have simplified a little the process for creating new gateways e.q. the gateway will be auto showed in **Setup->Settings->Payment Gateways**, encrypting fields, 1 unique function from each gateway library to call etc… but this will still require effort to get started.

Before start, make sure that you set [development mode](https://help.perfexcrm.com/faq/how-to-set-debug-mode/) in order to see any errors and functions/hooks deprecation warnings.

Let’s assume for this example your gateway name is **Example**

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Example\_gateway extends App\_gateway

{

    public function \_\_construct()

    {

        /\*\*

        \* Call App\_gateway \_\_construct function

        \*/

        parent::\_\_construct();

        /\*\*

         \* Gateway unique id - REQUIRED

\*

         \* \* The ID must be alphanumeric

         \* \* The filename (Example\_gateway.php) and the class name must contain the id as ID\_gateway

         \* \* In this case our id is "example"

         \* \* Filename will be Example\_gateway.php (first letter is uppercase)

         \* \* Class name will be Example\_gateway (first letter is uppercase)

         \*/

        $this->setId('example');

        /\*\*

         \* REQUIRED

         \* Gateway name

         \*/

        $this->setName('Example');

        /\*\*

         \* Add gateway settings

         \* You can add other settings here

\* to fit for your gateway requirements

         \*

         \* Currently only 3 field types are accepted for gateway

         \*

         \* 'type'=>'yes\_no'

         \* 'type'=>'input'

\* 'type'=>'textarea'

\*

         \*/

        $this->setSettings(array(

            array(

                'name' => 'api\_secret\_key',

                'encrypted' => true,

                'label' => 'API KEY',

                'type'=>'input',

            ),

            array(

                'name' => 'api\_publishable\_key',

                'label' => 'SECRET KEY',

                'type'=>'input'

            ),

            array(

                'name' => 'currencies',

                'label' => 'settings\_paymentmethod\_currencies',

                'default\_value' => 'USD,CAD'

            ),

        ));

    }

    /\*\*

     \* Each time a customer click PAY NOW button on the invoice HTML area, the script will process the payment via this function.

     \* You can show forms here, redirect to gateway website, redirect to Codeigniter controller etc..

     \* @param  array $data - Contains the total amount to pay and the invoice information

     \* @return mixed

     \*/

    public function process\_payment($data)

    {

        var\_dump($data);

        die;

    }

}

**There are comments on the functions and variables, you should spend some time to read them because they are important if you want to help you get started.**

Additionally, you can take a look at the other gateway example in **application/libraries/gateways** or **application/controllers/gateways**.

### **Excluding URL from CSRF**

If the module payment gateway is using webhooks and makes a POST request to a URL in Perfex CRM to notify about the payment updates, you will need to exclude this URL from CSRF in order for the request to pass.

You can follow [this](https://help.perfexcrm.com/working-with-forms/#exclude-url-from-csrf-v2-9-0) guide in order to achieve this.

# **Action Hooks**

/[Customization](https://help.perfexcrm.com/category/customization/) /Action Hooks

* March 16, 2017
* [Customization](https://help.perfexcrm.com/category/customization/)

Perfex CRM support various action hooks all over the code to give the buyers the best experience.

Create file with name [**my\_functions\_helper.php**](https://www.perfexcrm.com/wp-content/uploads/2016/08/my_functions_helper.zip) in application/helpers/. This file is reserved for all your functions. Here is how the [**my\_functions\_helper.php**](https://www.perfexcrm.com/wp-content/uploads/2016/08/my_functions_helper.zip) file template looks like /before you change it adding your code:  
  
<?php

/\* From version 1.2.0 my\_functions\_helper.php removed from the main files and need to be created when needed.

/\* Upload this file to application/helpers IF DONT EXISTS \*/

/\* Add your own functions here \*/

Before add any code in my\_functions\_helper.php, make sure that you set [development mode](https://help.perfexcrm.com/faq/how-to-set-debug-mode/) in order to see any errors and functions/hooks deprecation warnings.

#### **Version 2.3.0 and above.**

hooks()->add\_action($tag, $function\_to\_add, $priority = 10, $accepted\_args = 1);

hooks()->add\_filter($tag, $function\_to\_add, $priority = 10, $accepted\_args = 1);

hooks()->do\_action($tag, $arg = '');

hooks()->apply\_filters($tag, $value, $additionalParams);

The functions below are prior to version 2.3.0

**Add Action Hook**

add\_action('action\_function','your\_function\_callback');

Example action hook for **after\_client\_added**Note: this code should be inside my\_functions\_helper.php mentioned earlier.

**This is example where the action do not expect to return $data.**

function callback\_after\_client\_added\_insert\_to\_local\_database($id)

{

$clientid = $id;

// Do Staff with $clientid

}

add\_action('after\_client\_added','callback\_after\_client\_added\_insert\_to\_local\_database');

**Example action hook for before\_client\_added  
This is example where the action expect to return $data**

function callback\_before\_client\_added\_check\_company($data)

{

if($data['company'] == 'Test'){

$data['company'] = '';

}

return $data;

}

add\_action('before\_client\_added','callback\_before\_client\_added\_check\_company');