

<b>Intro</b>	<b>1</b>
📄 Disclaimer	1.1
📄 Requirements	1.2
📄 Installation	1.3
📄 Updates	1.4
<b>Settings</b>	<b>2</b>
📄 General Settings	2.1
📄 Advanced Settings	2.2
<b>Creating A New Form</b>	<b>3</b>
📄 Form Builder	3.1
<b>Form Settings</b>	<b>4</b>
📄 Form Settings	4.1
📄 General Settings	4.2
📄 Style Settings	4.3
📄 Confirmation Settings	4.4
📄 Double Opt-In Settings	4.5
📄 Notification Settings	4.6
📄 Integration Settings	4.7
📄 Math Expressions	4.8
📄 Payment Gateways	4.9
📄 Custom JavaScript Handlers	4.10
<b>Form Elements</b>	<b>5</b>
📄 General Info	5.1
📄 Text Field	5.2

 Email Field	5.3
 Textarea	5.4
 Select Box (Drop-down List)	5.5
 Checkbox	5.6
 Radio Button	5.7
 Multiselect	5.8
 Image Select	5.9
 Date Picker	5.10
 Time Picker	5.11
 File Upload	5.12
 Password	5.13
 Star Rating	5.14
 Hidden Field	5.15
 Button	5.16
 Custom HTML	5.17
 Link Button	5.18
 Column Layouts	5.19

# 1 INTRO

## 1.1 DISCLAIMER

---

We are constantly improving our script by adding new features and fixing bugs. Please keep the script updated and always use latest version. This documentation is related

to actual version of the plugin, so don't forget to check it from time to time. **This documentation is under development.**

- Initial release: January 15th, 2019
- Latest update: April 21st, 2019
- Version: 1.10

## 1.2 REQUIREMENTS

---

Your server must match the following requirements. All modern servers already do it.

- PHP version 5.5 or higher.
- MySQL version 5.0 or higher.

## 1.3 INSTALLATION

---

We created Wizard which helps to install script properly.

1. Unpack downloaded archive and upload files to your server.
2. Open it in your browser – Wizard will be launched automatically.
3. Wizard ask for couple questions. Nothing special – MySQL parameters and admin credentials to access Admin Panel.
4. Once finished, go to Dashboard and click "Green Forms" box under "Installed Plugins" section. It activates the plugin.
5. Once activated, plugin adds relevant item to Left Side Menu.
6. Find "How To Use" submenu. It explains how to embed plugin into your wesbite – nothing difficult, just paste couple JS/HTML-snippets.

## 1.4 UPDATES

---

Updating the script is a manual procedure. It's pretty simple. Don't worry.

1. Download latest version of the script from CodeCanyon.
2. Overwrite existing files on your server, except file `/inc/config.php` and folder `/content/data/`.

You won't lose forms and settings. They are stored in MySQL database.

## 2 SETTINGS

### 2.1 GENERAL SETTINGS

---

General Settings can be reached through menu "Green Forms >> Settings".

#### Mailing Settings

All email messages sent by script use the following parameters for "FROM:" header.

You also can override these parameters for certain form, if required.

- **Sender Name.** All messages from the script are sent using this name as "FROM:" header value.
- **Sender Email.** All messages from the script are sent using this email address as "FROM:" header value.

#### Miscellaneous

- **GA Tracking.** Enable this option to send form submission event to Google Analytics. Google Analytics must be installed on your website. If you use Google Tags Manager, please configure it properly to accept events.
- **Font Awesome.** Enable this option if you want to use full set of Font Awesome icons. Otherwise, limited set of icons is used (for better performance). If your site already loads Font Awesome, you can enable relevant option to avoid double loading of the same library.
- **Air Datepicker plugin.** Enable this option if you want to use nice datepicker (Air Datepicker) with date and time fields. If your site already loads Air Datepicker, you can enable relevant option to avoid double loading of the same plugin.
- **jQuery Mask plugin.** For some type of form elements you can use input masks. This feature is driven by jQuery Mask plugin. Enabling it, allows you to set certain input mask for input fields. If your site already loads jQuery Mask plugin, you can enable relevant option to avoid double loading of the same library.
- **JavaScript Expression Parser.** Enable this feature if you want to show result of math expressions in real-time. If your site already loads JavaScript Expression Parser, you can enable relevant option to avoid double loading of the same plugin.

#### AWeber Connection

This section appears if you activated AWeber Integration module (read Plugin Modules paragraph regarding enabling/disabling modules). In this section you can connect website with your AWeber account. After successful connection, you can configure integration on Form Settings window, Integrations tab.

## Zoho CRM Connection


This section appears if you activated Zoho CRM Integration module (read Plugin Modules paragraph regarding enabling/disabling modules). In this section you can connect website with your Zoho CRM account. After successful connection, you can configure integration on Form Settings window, Integrations tab.

## Item Purchase Code

To activate your license please enter Item Purchase Code. Where can I find my Purchase Code?

## 2.2 ADVANCED SETTINGS

---

Advanced plugin settings can be reached through menu "Green Forms >> Settings", tab .

### Plugin Modules

Plugin has modular architecture. It easily allows you to enable/disable certain functionality. For example, if you need MailChimp integration or accept payments via PayPal, just activate appropriate module and configure it on form editor. Actual version of the plugin has the following default marketing, newsletter and CRM modules:

- **3rd party HTML form.** Turn this module on if you want to re-submit form data as a part of 3rd party HTML form. After activation you can configure integration on form editor.
- **ActiveCampaign.** Turn this module on if you want to submit form data to ActiveCampaign. This module uses ActiveCampaign API. After activation you can configure integration on form editor.
- **ActiveTrail.** Turn this module on if you want to submit form data to ActiveTrail. This module uses ActiveTrail API. After activation you can configure integration on form editor.
- **AgileCRM.** Turn this module on if you want to submit form data to AgileCRM. This module uses AgileCRM API. After activation you can configure integration on form editor.

- **AvangEmail.** Turn this module on if you want to submit form data to AvangEmail. This module uses AvangEmail API. After activation you can configure integration on form editor.
- **AWeber.** Turn this module on if you want to submit form data to AWeber. This module uses AWeber API. After activation connect to AWeber on General Settings page and configure integration on form editor.
- **Campaign Monitor.** Turn this module on if you want to submit form data to Campaign Monitor. This module uses Campaign Monitor API. After activation you can configure integration on form editor.
- **Conversio.** Turn this module on if you want to submit form data to Conversio. This module uses Conversio API. After activation you can configure integration on form editor.
- **FreshMail.** Turn this module on if you want to submit form data to FreshMail. This module uses FreshMail API. After activation you can configure integration on form editor.
- **GetResponse.** Turn this module on if you want to submit form data to GetResponse. This module uses GetResponse API. After activation you can configure integration on form editor.
- **Jetpack Subscriptions.** Turn this module on if you want to submit form data to Jetpack Subscriptions. (This module is available for WordPress version only). After activation you can configure integration on form editor.
- **Mad Mimi.** Turn this module on if you want to submit form data to Mad Mimi. This module uses Mad Mimi API. After activation you can configure integration on form editor.
- **Mailautic.** Turn this module on if you want to submit form data to Mailautic. This module uses Mailautic API. After activation you can configure integration on form editor.
- **MailChimp.** Turn this module on if you want to submit form data to MailChimp. This module uses MailChimp API. After activation you can configure integration on form editor.
- **MailerLite.** Turn this module on if you want to submit form data to MailerLite. This module uses MailerLite API. After activation you can configure integration on form editor.
- **MailPoet.** Turn this module on if you want to submit form data to MailPoet. (This module is available for WordPress version only). After activation you can configure integration on form editor.
- **Mailster.** Turn this module on if you want to submit form data to Mailster. (This module is available for WordPress version only). After activation you can configure integration on form editor.

- **MailWizz.** Turn this module on if you want to submit form data to MailWizz. This module uses MailWizz API. After activation you can configure integration on form editor.
- **Moosend.** Turn this module on if you want to submit form data to Moosend. This module uses Moosend API. After activation you can configure integration on form editor.
- **Mumara.** Turn this module on if you want to submit form data to Mumara. This module uses Mumara API. After activation you can configure integration on form editor.
- **Omnisend.** Turn this module on if you want to submit form data to Omnisend. This module uses Omnisend API. After activation you can configure integration on form editor.
- **Ontraport.** Turn this module on if you want to submit form data to Ontraport. This module uses Ontraport API. After activation you can configure integration on form editor.
- **Rapidmail.** Turn this module on if you want to submit form data to Rapidmail. This module uses Rapidmail API. After activation you can configure integration on form editor.
- **Zoho CRM.** Turn this module on if you want to submit form data to Zoho CRM. This module uses Zoho CRM API. After activation connect to Zoho CRM on General Settings page and configure integration on form editor.

Actual version of the plugin has the following payment provider modules:

- **Blockchain.** Turn this module on if you want to accept bitcoin payments via Blockchain. After activation you can configure payment gateway on form editor.
- **InterKassa.** Turn this module on if you want to accept payments via InterKassa. After activation you can configure payment gateway on form editor.
- **PayPal.** Turn this module on if you want to accept payments via PayPal. After activation you can configure payment gateway on form editor.
- **Perfect Money.** Turn this module on if you want to accept payments via Perfect Money. After activation you can configure payment gateway on form editor.
- **Skrill.** Turn this module on if you want to accept payments via Skrill. After activation you can configure payment gateway on form editor.
- **Stripe.** Turn this module on if you want to accept credit/debit cards via Stripe. After activation you can configure payment gateway on form editor.
- **Yandex.Money.** Turn this module on if you want to accept payments via Yandex.Money. After activation you can configure payment gateway on form editor.

## Miscellaneous

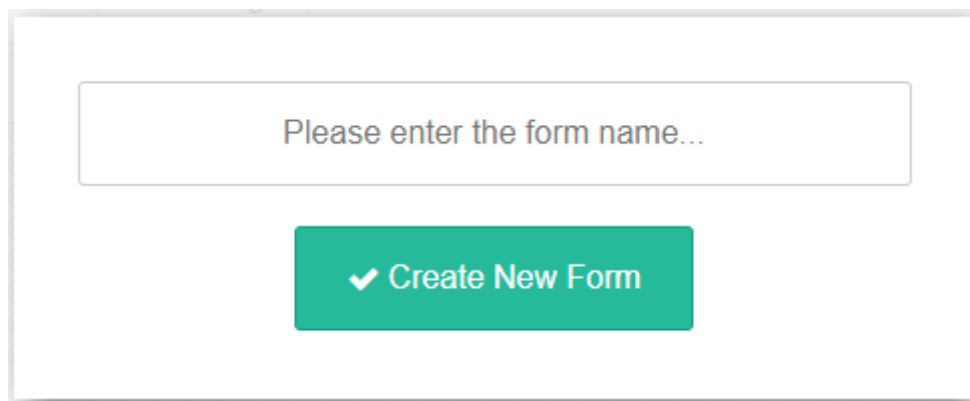
- **Minification.** Turn this feature on if you want to load minified version of CSS and JS files.



## 3 CREATING A NEW FORM

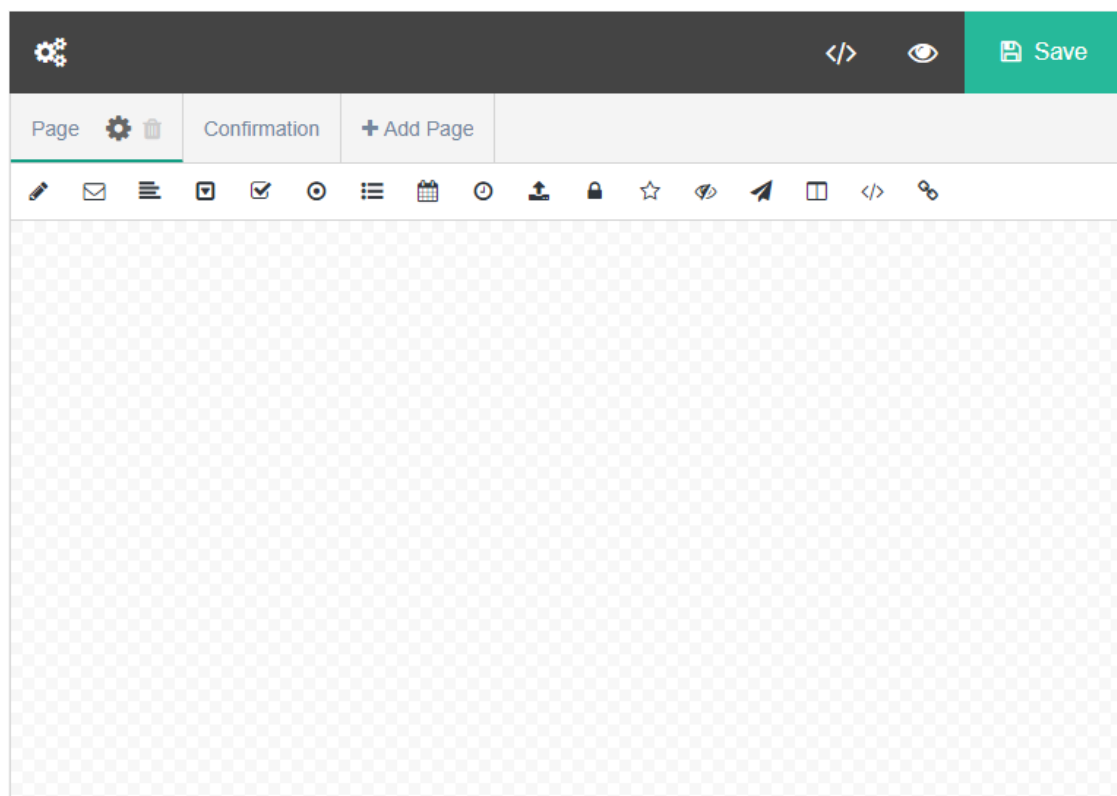
### 3.1 FORM BUILDER

Plugin goes with easy-to-use drag-n-drop form builder. To start creating a new form just click "Create From" in Left Side Menu. You will be prompted to enter a name for the new form, once you've done so click "Create New Form" button to continue.

A screenshot of a dialog box for creating a new form. It features a light gray background. In the center, there is a white rectangular input field with the placeholder text "Please enter the form name...". Below the input field is a green rectangular button with a white checkmark icon and the text "Create New Form".





You can then start adding elements and configuring the form.

Form Builder consists of several tool bars (top toolbar, pages/steps toolbar, form elements toolbar) and "construction area".




## Top toolbar

Top toolbar contains the following buttons (from left to right):


- **Form Settings** (  ). This button opens "Form Settings" window. It has hundreds adjustable parameters related to general form functionality, design, notifications, integrations, etc. All of them are described in this documentation. Just scroll down to appropriate section.
- **How To Use** (  ). This button opens "How To Use" window. There you can find all possible ways of using of the form (inline, popups, etc.).
- **Preview** (  ). Click this button if you want preview the form before publishing it on the website. *Important! Notifications and Integrations don't work in preview mode. Preview mode is used to check form design only.*
- **Save** (  Save ). Click this button to save the changes.

## Pages/steps Toolbar

This bar is used to build multi-pages (multi-steps) forms. By clicking button













 **Add Page** you can add as many pages/steps as you want.







There is one special page/step which always exists from the beginning -

 **Confirmation** . The content of this page/step is displayed when users successfully submit the form. *Important! This is not the only way to inform users about successful form submission. For more options please visit the chapter below.*

## Form Elements Toolbar

This toolbar contains form element icons. If you want to add some element to the form (for example, input field or button or something else), just click relevant icon and new element will be added. Then you can move new element into desired position and configure it. Following elements can be added to the form:

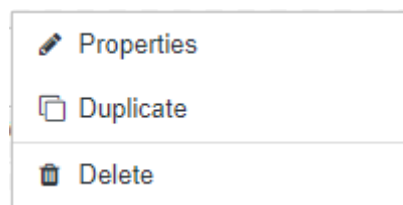
-  - Text Input Field
-  - Email Input Field
-  - Textarea
-  - Select Box (Drop-down List)
-  - Checkbox
-  - Radio Button
-  - Multiselect List
-  - Image Select
-  - Date Picker
-  - Time Picker
-  - File Upload
-  - Password

-  - Star Rating
-  - Hidden Field
-  - Submit Button
-  - Column Layouts
-  - HTML Content
-  - Link Button

The properties of each element are described in details in chapter "Form Elements".

## Construction Area

This is an area of Form Builder where you create the form. If you want to add some element to the form (for example, input field or button or something else), just click relevant icon on Form Elements Toolbar and new element will be added. Use mouse to move element into desired position. Each element has context menu, just do right click over it.




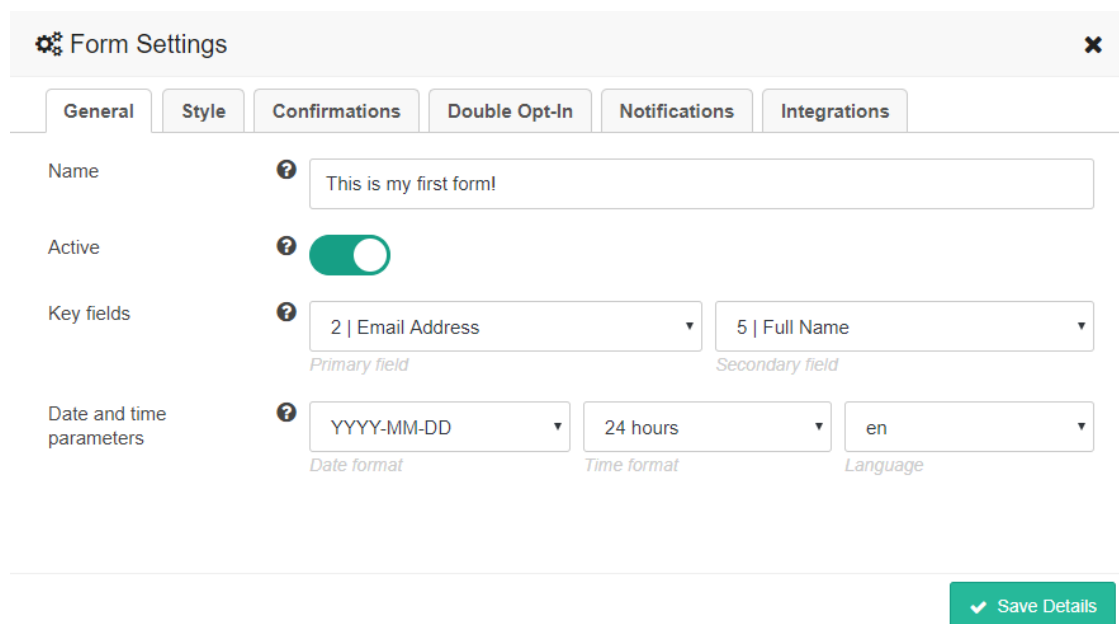
Context Menu contains the following items:

- **Properties.** Click it to open "Element Properties" window. Each form element has dozens properties that you can change. For more details regarding properties, please read appropriate chapter.
- **Duplicate.** Create exact copy (including all properties) of the element.
- **Delete.** Delete the element.


## 4 FORM SETTINGS

### 4.1 FORM SETTINGS

The Form Settings window appears when you click  button on Top Toolbar. It has hundreds adjustable parameters related to general form functionality, design, notifications, integrations, etc. All of them are described in this documentation. Form Settings Window looks like below:



### 4.2 GENERAL SETTINGS


Click  button on Top Toolbar and select "General" tab. It contains the following general form settings:

- **Name.** The name is used for your convenience. It helps to identify the form on the site.
- **Active.** Inactive forms will not appear on the site even if you inserted into pages. It allows to disable the form globally.
- **Key fields.** This is 2 main fields of the form. Values of these fields are displayed on Log page in relevant columns. It helps to identify submitted form record without opening its full details.

- **Date and time parameters.** Choose the date/ time formats and language for datetimepicker. It is used for "date" and "time" fields.

## 4.3 STYLE SETTINGS

---

Click  button on Top Toolbar and select **Style** tab. It contains a lot of style settings grouped by categories.

### Global Style Settings

This category contains global parameters of the form:

- **Text style.** Adjust the style (font, color, size, etc.) of the text that appears in your form.
- **Wrapper background.** Adjust the background style of the form. You can use separate settings for inline and for popup modes.
- **Wrapper border.** Adjust the border style of the form. You can use separate settings for inline and for popup modes.
- **Wrapper shadow.** Adjust the shadow style of the form. You can use separate settings for inline and for popup modes.
- **Padding.** Adjust the padding of the form. You can use separate settings for inline and for popup modes.
- **Overlay color.** Adjust the overlay color when the form is used in popup mode.
- **Close icon colors.** Adjust colors of the close icon when the form is used in popup mode.
- **Spinner colors.** Adjust colors of the spinner when the form is used in popup mode.
- **Form width.** Specify the maximum form width and its alignment. Leave this field empty to set maximum form width as 100%.
- **Element spacing.** Specify the spacing between form elements.
- **Responsiveness.** At what form width should column layouts be stacked.

### Label Style Settings

Here you can configure the style of labels and descriptions. To understand what is label and what is description please look at image below. Green text is label, red text is description.



- **Label text style.** Adjust the style (font, color, size, etc.) of labels. These parameters override "Text style" parameters from "Global Style Settings" category.
- **Label position.** Specify where to display the label relative to the field.
- **Description text style.** Adjust the style (font, color, size, etc.) of descriptions. These parameters override "Text style" parameters from "Global Style Settings" category.
- **Description position.** Specify where to display the description relative to the field.

## Input Field Style Settings

This category contains parameters to adjust the style of input fields.

- **Input size.** Choose the size of input fields.
- **Icon style.** Adjust the style (size, color, position) of input field icons.
- **Textarea height.** Set the height of textarea fields.
- **Input text.** Adjust the text style (font, color, size, etc.) of input fields. You can do it for different input states (normal, hover, focus). These parameters override "Text style" parameters from "Global Style Settings" category.
- **Input background.** Adjust the background style of input fields. You can do it for different input states (normal, hover, focus).
- **Input border.** Adjust the border style of input fields. You can do it for different input states (normal, hover, focus).
- **Input shadow.** Adjust the shadow style of input fields. You can do it for different input states (normal, hover, focus).
- **Checkbox and radio style.** Choose how to display checkbox and radio button fields and their captions.
- **Checkbox view.** Choose the checkbox view.
- **Radio button view.** Choose the radio button view.
- **Image Select style.** Adjust image alignment and effect.
- **Image border.** Adjust the border style of images. You can do it for different states (normal, hover, selected).
- **Image shadow.** Adjust the shadow style of images. You can do it for different states (normal, hover, selected).

## Button Style Settings

This category contains parameters to adjust the style of buttons.

- **Button style.** Adjust the button size and position.
- **Button text.** Adjust the text style (font, color, size, etc.) of buttons. You can do it for different button states (normal, hover, active). These parameters override "Text style" parameters from "Global Style Settings" category.

- **Button background.** Adjust the background style of buttons. You can do it for different button states (normal, hover, active).
- **Button border.** Adjust the border style of buttons. You can do it for different button states (normal, hover, active).
- **Button shadow.** Adjust the shadow style of buttons. You can do it for different button states (normal, hover, active).

## Error Bubble Style Settings


If user submit wrong data through certain input field, the error bubble is displayed for this field. In this category you can configure the view of the bubble.

- **Bubble background.** Adjust the background of error bubbles.
- **Error text style.** Adjust the text style (font, color, size, etc.) of errors. These parameters override "Text style" parameters from "Global Style Settings" category.

## 4.4 CONFIRMATION SETTINGS

---

Confirmation is an action that is executed on front-side when form successfully submitted. By default it is a displaying of special page/step which is called "Confirmation". You can see and edit this page by clicking [Confirmation](#) on Pages/steps Toolbar.

If you need more specific action, click  button on Top Toolbar and select "Confirmations" tab. There you can customize confirmation and use conditional logic. There are several action to select:

- **Display Confirmation page.** It is the same action as by default, but you also can apply conditional logic to this action and reset form to default state, if required.
- **Display Confirmation page and redirect to certain URL.** The same as above, but after specified number of seconds user will be redirected to specific URL.
- **Display Confirmation page and request payment.** The same as above, but after specified number of seconds user will be requested to complete payment.
- **Display Message.** The message is displayed below the form. It stays visible during specified number of seconds and then disappears.
- **Display Message and redirect to certain URL.** The message is displayed below the form. It stays visible during specified number of seconds and then user will be redirected to specific URL.
- **Display Message and request payment.** The message is displayed below the form. It stays visible during specified number of seconds and then user will be requested to complete payment.


- **Redirect to certain URL.** User is redirected to specific URL immediately.
- **Request payment.** User is requested to complete payment immediately.

You can apply conditional logic to each action. This means that this action will be performed only if the data submitted by the user meets certain criteria.

*Important! If several confirmations match form conditions, the first one (higher priority) will be applied. So, you can use mouse to re-order multiple confirmations.*

## 4.5 DOUBLE OPT-IN SETTINGS

---

If you want users to confirm submitted data click  button on Top Toolbar and select **Double Opt-In** tab and enable this feature. If enabled, the plugin sends email message with confirmation link to certain email address (submitted by user). When confirmation link clicked, relevant record is marked as "confirmed".


**Double Opt-In** tab has the following parameters to configure:

- **Enable.** Enable or disable double opt-in.
- **Recipient.** The email address to which confirmation link will be sent to.
- **Subject.** The subject of the email message.
- **Message.** The content of the email message. It is mandatory to include `{{confirmation-url}}` into content.
- **From.** The "From" address and name. The email address and name set here will be shown as the sender of the email.
- **Thanksgiving message.** This message is displayed when users successfully confirmed their e-mail addresses.
- **Thanksgiving URL.** This is alternate way of thanksgiving message. After confirmation users are redirected to this URL.

*Important! This feature is disabled if the form requests users to complete the payment.*

## 4.6 NOTIFICATION SETTINGS

---

The notification, welcome, thanksgiving or whatever email can be sent to user, administrator or any other email address. To enable this feature just click  button on Top Toolbar, select **Notifications** tab and create as many notification as you need. You can customize these emails and use conditional logic.

Each notification has the following parameters to configure:

- **Name.** The name of the notification. It is used for your convenience.
- **Enabled.** You can stop this notification being sent by turning this off.




- **Send.** Specify when notification must be sent: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **Recipient.** Add email addresses (comma-separated) to which this email will be sent to.
- **Subject.** The subject of the email message.
- **Message.** The content of the email message.
- **Attachments.** Select files that you want to attach to the email message.
- **Reply-To.** Add a "Reply-To" email address. If not set, replying to the email will reply to the "From" address.
- **From.** Sets the "From" address and name. The email address and name set here will be shown as the sender of the email.
- **Conditional logic.** If enabled, you can create rules to enable this notification depending on the values of input fields.

You can apply conditional logic to each notification. This means that email message will be sent only if the data submitted by the user meets certain criteria.

## 4.7 INTEGRATION SETTINGS

---

Form data submitted by user can be sent to 3rd party services (such as MailChimp, AWeber, GetResponse, etc.). To enable this feature just click  button on Top Toolbar, select [Integrations](#) tab and create as many integrations as you need. You can customize integrations and use conditional logic.

*Important! If you do not see your marketing/CRM provider, make sure that you enabled appropriate integration module on [Advanced Settings](#) page.*



Plugin has integration modules with following marketing/CRM providers:

- 3rd party HTML form
- ActiveCampaign
- ActiveTrail
- AgileCRM
- AvangEmail
- AWeber
- Campaign Monitor
- Conversio
- FreshMail
- GetResponse
- Jetpack Subscriptions
- Mad Mimi

- Mailautic
- MailChimp
- MailerLite
- MailPoet
- Mailster
- MailWizz
- Moosend
- Mumara
- Omnisend
- Ontraport
- Rapidmail
- Zoho CRM

If you don't see your provider in a list, just drop request and we will add it. 🙄

## HTML form

To enable integration with 3rd party HTML form, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate HTML Form Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ HTML Form](#) button. New integration will be added. Click  to configure integration.



It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **HTML form.** Copy-paste HTML-code of 3rd party form and connect it.  
*Important! It must be pure HTML-form with <form> tag, but not javascript/iframe-driven form.*
- **Fields.** Map form fields to 3rd party form fields.
- **Client side.** Enable this option if you want form to be submitted from client side. Otherwise, it is submitted from server side. Please read paragraph below regarding using this feature.
- **Target.** Select where to display the response that is received after submitting the form. This feature is only available, "Client side" enabled.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

Pay attention to some combination of parameters. Some of them won't work.

- "Execute" is "after successfully completed payment" and "Client side" is enabled. Client side is not involved into notification about successful payment received from payment provider.
- "Execute" is "after non-completed payment" and "Client side" is enabled. Client side is not involved into notification about non-completed payment received from payment provider.



## ActiveCampaign

To enable integration with ActiveCampaign, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate ActiveCampaign Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ ActiveCampaign](#) button. New ActiveCampaign integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API URL.** Enter your ActiveCampaign API URL. To get API URL please go to your ActiveCampaign Account >> Settings >> Developer.
- **API Key.** Enter your ActiveCampaign API Key. To get API Key please go to your ActiveCampaign Account >> Settings >> Developer.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to ActiveCampaign fields.
- **Tags.** Enter comma-separated list of tags.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## ActiveTrail



To enable integration with ActiveTrail, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate ActiveTrail Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ ActiveTrail](#) button. New ActiveTrail integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.

- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **Access Token.** Enter your ActiveTrail Access Token. Create it on Settings page in ActiveTrail account.
- **Fields.** Map form fields to ActiveTrail fields.
- **Groups.** Select groups.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.



## AgileCRM

To enable integration with AgileCRM, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate AgileCRM Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ AgileCRM](#) button. New AgileCRM integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **Site URL.** Enter unique website address of your account. Usually it looks like `https://SITE-NAME.agilecrm.com`.
- **Email.** Enter email address of your AgileCRM account (i.e. email address that you used to create AgileCRM account).
- **API Key.** Enter your AgileCRM REST API Key. Find it in your AgileCRM account, click "Admin Settings" and "Developers & API".
- **Campaign ID.** Select Campaign ID where data submitted to.
- **System Properties.** Map form fields to AgileCRM system properties.
- **Custom Properties.** Configure AgileCRM custom properties and map form fields to them.
- **Tags.** If you want to tag contact with tags, drop them here (comma-separated string).
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.



## AvangEmail

To enable integration with AvangEmail, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate AvangEmail Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ AvangEmail](#) button. New AvangEmail integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **Public Key.** Enter your AvangEmail API Public Key. Find it here.
- **Private Key.** Enter your AvangEmail API Private Key. Find it here.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to AvangEmail fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.


## AWeber

To enable integration with AWeber, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate AWeber Integration module](#). Then go to [General Settings](#) page, find [AWeber Connection](#) section and connect plugin to your AWeber account. After successful connection go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ AWeber](#) button. New AWeber integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to AWeber fields.
- **Tags.** Enter comma-separated list of tags applied to the contact.
- **Notes.** Enter notes applied to the contact (max 60 sybmols).
- **Ad Tracking.** Enter your Ad Tracking info applied to the contact.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.



## Campaign Monitor

To enable integration with Campaign Monitor, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Campaign Monitor Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Campaign Monitor](#) button. New Campaign Monitor integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API Key.** Enter your Campaign Monitor API Key. Find Campaign Monitor API Key on Account Settings page when logged into your Campaign Monitor account.
- **Client ID.** Select Client ID.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to Campaign Monitor fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## Conversio



To enable integration with Conversio, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Conversio Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Conversio](#) button. New Conversio integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API Key.** Enter your Conversio API Key. Find Conversio API Key on Profile page.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to Conversio fields.
- **Opt-In Text.** What opt-in text was shown to the subscriber. This is required for GDPR compliance.

- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.



## FreshMail

To enable integration with FreshMail, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate FreshMail Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ FreshMail](#) button. New FreshMail integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API Key.** Enter your FreshMail API Key. Find it on Integration page in FreshMail dashboard.
- **API Secret.** Enter your FreshMail API Secret. Find it on Integration page in FreshMail dashboard.
- **List ID.** Select List ID where data submitted to.
- **Fields.** Map form fields to FreshMail fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## GetResponse

To enable integration with GetResponse, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate GetResponse Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ GetResponse](#) button. New GetResponse integration will be added. Click  to configure integration.



It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API Key.** Enter your GetResponse API Key. Find it here.
- **Campaign ID.** Select Campaign ID where data submitted to.
- **Fields.** Map form fields to GetResponse fields.

- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## Jetpack Subscriptions



*Important! Make sure that you installed and activated Jetpack plugin.*

To enable integration with Jetpack Subscriptions, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Jetpack Subscriptions Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Jetpack Subscriptions](#) button. New Jetpack Subscriptions integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **Fields.** Map certain form field to Jetpack email field.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## Mad Mimi

To enable integration with Mad Mimi, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Mad Mimi Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Mad Mimi](#) button. New Mad Mimi integration will be added. Click  to configure integration.



It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **Username.** Enter Mad Mimi username or email.
- **API Key.** Enter Mad Mimi API Key. Find it on Account page (right side column) in your Mad Mimi account.
- **List ID.** Select List ID where data submitted to.
- **Fields.** Map form fields to Mad Mimi fields.
- **Custom Fields.** Map form fields to Mad Mimi custom fields.



- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.



## Mailautic

To enable integration with Mailautic, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Mailautic Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Mailautic](#) button. New Mailautic integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **Public Key.** Enter your Mailautic API Public Key. Find it here.
- **Private Key.** Enter your Mailautic API Private Key. Find it here.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to Mailautic fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## MailChimp



To enable integration with MailChimp, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate MailChimp Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ MailChimp](#) button. New MailChimp integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API Key.** Enter your MailChimp API Key. Find it here.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to MailChimp fields.
- **Groups.** Select groups.

- **Tags.** Comma-separated list of tags. Tagging lets you bring your own contact structure into Mailchimp and label contacts based on data only you know about them.
- **Double opt-in.** Control whether a double opt-in confirmation message is sent.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## MailerLite



To enable integration with MailerLite, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate MailerLite Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ MailerLite](#) button. New MailerLite integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API Key.** Enter your MailerLite API Key. Find it on Developer API page.
- **Group ID.** Select group ID where data submitted to.
- **Fields.** Map form fields to MailChimp fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## MailPoet

*Important! Make sure that you installed and activated MailPoet plugin.*

To enable integration with MailPoet, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate MailPoet Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ MailPoet](#) button. New MailPoet integration will be added. Click  to configure integration.



It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **List ID.** Select list ID where data submitted to.

- **Fields.** Map form fields to MailPoet fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## Mailster



*Important! Make sure that you installed and activated Mailster plugin.*

To enable integration with Mailster, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Mailster Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Mailster](#) button. New Mailster integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to Mailster fields.
- **Double opt-in.** Control whether a double opt-in confirmation message is sent.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## Mailwizz



To enable integration with Mailwizz, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Mailwizz Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Mailwizz](#) button. New Mailwizz integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API URL.** Enter your MailWizz API URL. If the MailWizz powered website does not use clean urls, make sure your API URL has the index.php part of url included.
- **Public Key.** Enter your MailWizz API Public Key. You can generate it in MailWizz customer area.

- **Private Key.** Enter your MailWizz API Private Key. You can generate it in MailWizz customer area.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to Mailwizz fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.



## Moosend

To enable integration with Moosend, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Moosend Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Moosend](#) button. New Moosend integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API Key.** Enter your Moosend API Key. You can get it on the settings page in your Moosend account.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to Moosend fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## Mumara



To enable integration with Mumara, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Mumara Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Mumara](#) button. New Mumara integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **URL.** Enter your Mumara URL.

- **API Token.** Enter your Mumara API Token. You can generate it in Mumara account (Settings >> API Key).
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to Mumara fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.



## Omnisend

To enable integration with Omnisend, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Omnisend Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Omnisend](#) button. New Omnisend integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API Key.** Enter your Omnisend API Key. Find it here.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to Omnisend fields.
- **Custom Fields.** Map form fields to Omnisend custom fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## Ontraport



To enable integration with Ontraport, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Ontraport Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Ontraport](#) button. New Ontraport integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.

- **App ID.** Enter your Ontraport App ID. Find your Ontraport App ID in Administration Settings.
- **API Key.** Enter your Ontraport API Key. Find your Ontraport API Key in Administration Settings.
- **Tags.** Select tags for submitted data.
- **Sequences.** Select sequences for submitted data.
- **Fields.** Map form fields to Ontraport fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.



## Rapidmail

To enable integration with Rapidmail, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Rapidmail Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Rapidmail](#) button. New Rapidmail integration will be added. Click  to configure integration.

It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.
- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **API Username.** Enter your Rapidmail API Username. Find it here.
- **API Password.** Enter your Rapidmail API Password. Find it here.
- **List ID.** Select list ID where data submitted to.
- **Fields.** Map form fields to Rapidmail fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.

## Zoho CRM

To enable integration with Zoho CRM, make sure that you activated appropriate integration module. Go to [Advanced Settings](#) page and turn on [Activate Zoho CRM Integration module](#). Then go to [General Settings](#) page, find [Zoho CRM Connection](#) section and connect plugin to your Zoho CRM account. After successful connection go to form editor, click  button on Top Toolbar, select [Integrations](#) tab and click [+ Zoho CRM](#) button. New Zoho CRM integration will be added. Click  to configure integration.


It has the following parameters:

- **Name.** The name of the integration. It is used for your convenience.

- **Enabled.** You can disable this integration by turning this off.
- **Execute.** You can specify when integration must be executed: after successful form submission, when user confirmed submitted data using double opt-in feature, after successfully completed payment or after non-completed payment.
- **Fields.** Map form fields to Zoho CRM fields.
- **Conditional logic.** If enabled, you can create rules to enable this integration depending on the values of input fields.


## 4.8 MATH EXPRESSIONS

---

The plugin allows to use basic math expressions along the form. To create math expression just click  button on Top Toolbar, select **Advanced** tab, choose **Math Expressions** and create as many math expressions as you need.


Each math expression has the following parameters to configure:

- **ID.** The unique ID of the expression. You can't change it.
- **Name.** The name of the expression. It is used for your convenience.
- **Expression.** Type math expression here. Use basic arithmetic operators: -, +, \*, /.
- **Default.** This value is used if expression can not be calculated (for example, in case of division by zero, typos, missed variables, non-numeric values, etc.).
- **Decimal digits.** Specify how many decimal digits the result must have.

Each Math Expression has its own unique shortcode, so you can insert it into different part of the form. For example, you can create new form element "Custom HTML", go to its properties and find property "HTML" under "Basic" tab. On left side hover mouse over  to see all available fields and math expressions. Clicking desired math expression will add its shortcode into HTML content.

## 4.9 PAYMENT GATEWAYS

---

After successful form submission user can be requested to pay some amount via certain payment gateway. To create and configure payment gateway just click  button on Top Toolbar, select **Advanced** tab, click **Payment Gateways** and create as many gateways as you need.

*Important! If you do not see your payment provider, make sure that you enabled appropriate module on **Advanced Settings** page.*



Plugin has integration modules with following payment providers:

- Blockchain
- InterKassa
- PayPal
- Perfect Money
- Skrill
- Stripe
- Yandex.Money

If you still don't see your provider in a list, just drop request and we will consider adding it (but not promise).

When payment gateway created, go to [Confirmations](#) tab, create confirmation of one of the following types: "Display Confirmation page and request payment", "Display Message and request payment" or "Request payment" and select desired gateway.



## Blockchain

Before creating payment gateway driven by Blockchain (accept bitcoins), make sure that you activated appropriate module. Go to [Advanced Settings](#) page and turn on [Activate Blockchain Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Advanced](#) tab, click [Payment Gateways](#) and click [+ Blockchain](#) button. New gateway based on Blockchain will be added. Click  to configure it.

It has the following parameters:

- **Name.** The name of the payment gateway. It is used for your convenience.
- **Receive Payments API Key.** Enter a valid Receive Payments API Key. You cannot use the standard blockchain wallet API key for Receive Payments V2, and vice versa. Please apply for an API Key at <https://api.blockchain.info/v2/apikey/request/>.
- **xPub.** Enter Extended Public Key (xPub). Find it in your Blockchain account: Settings >> Addresses >> Manage >> More Options >> Show xPub.
- **Secret.** Enter random alphanumeric string.
- **Confirmations.** Set the number of confirmations of transaction.
- **Amount.** Set amount to pay and currency.

## InterKassa

Before creating payment gateway driven by InterKassa, make sure that you activated appropriate module. Go to [Advanced Settings](#) page and turn on [Activate InterKassa Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Advanced](#) tab, click [Payment Gateways](#) and click [+ InterKassa](#) button. New gateway based on InterKassa will be added. Click  to



configure it.



It has the following parameters:

- **Name.** The name of the payment gateway. It is used for your convenience.
- **Checkout ID.** Enter a valid InterKassa Checkout ID. All payments are sent to this account. *Important! Make sure that feature "Allow override in request" for "Interaction URL" is activated. Find it on Checkout Settings page, Interface tab in your InterKassa account.*
- **Sign Key.** Enter Sign Key. Find it on Checkout Settings page, Security tab in your InterKassa account.
- **Sign Algorithm.** Enter Sign Algorithm. It must be the same as on Checkout Settings page, Security tab in your InterKassa account.
- **Item name.** Enter the item name.
- **Amount.** Set amount to pay and currency.
- **Successful payment URL.** All payers are redirected to this URL after successful payment.
- **Cancel URL.** All payers are redirected to this URL in case of failed/cancelled payment.

## PayPal

Before creating payment gateway driven by PayPal, make sure that you activated appropriate module. Go to [Advanced Settings](#) page and turn on

[Activate PayPal Integration module](#).

Then go to form editor, click  button on Top Toolbar, select [Advanced](#) tab, click [Payment Gateways](#) and click [+ PayPal](#) button. New gateway based on PayPal will be added. Click  to configure it.

It has the following parameters:


- **Name.** The name of the payment gateway. It is used for your convenience.
- **PayPal ID.** Enter a valid PayPal email or Merchant ID. All payments are sent to this account.
- **Item name.** Enter the item name.
- **Amount.** Set amount to pay and currency.
- **Successful payment URL.** All payers are redirected to this URL after successful payment.
- **Cancel URL.** All payers are redirected to this URL in case of failed/cancelled payment.

## Perfect Money

Before creating payment gateway driven by Perfect Money, make sure that you activated appropriate module. Go to [Advanced Settings](#) page and turn on

[Activate Perfect Money Integration module](#).



Then go to form editor, click .

button on Top Toolbar, select **Advanced** tab, click **Payment Gateways** and click **+ Perfect Money** button. New gateway based on Perfect Money will be added. Click  to configure it.

It has the following parameters:

- **Name.** The name of the payment gateway. It is used for your convenience.
- **Payee Account.** The merchant Perfect Money account to which the payment is to be made.
- **Payee Name.** The name the merchant wishes to have displayed as the Payee on the Perfect Money payment form.
- **Alternate Passphrase.** Enter valid Alternate Passphrase. Find it on Settings page.
- **Item name.** Enter the item name.
- **Amount.** Set amount to pay and currency.
- **Successful payment URL.** All payers are redirected to this URL after successful payment.
- **Cancel URL.** All payers are redirected to this URL in case of failed/cancelled payment.

## Skrill



Before creating payment gateway driven by Skrill, make sure that you activated appropriate module. Go to **Advanced Settings** page and turn on **Activate Skrill Integration module**. Then go to form editor, click  button on Top Toolbar, select **Advanced** tab, click **Payment Gateways** and click **+ Skrill** button. New gateway based on Skrill will be added. Click  to configure it.

It has the following parameters:

- **Name.** The name of the payment gateway. It is used for your convenience.
- **Merchant email.** Enter a valid email address of your Skrill merchant account.
- **Secret word.** Please enter a valid secret word. Find it on Developer Settings page in your Skrill account.
- **Item name.** Enter the item name.
- **Amount.** Set amount to pay and currency.
- **Successful payment URL.** All payers are redirected to this URL after successful payment.
- **Cancel URL.** All payers are redirected to this URL in case of failed/cancelled payment.

## Stripe

Before creating payment gateway driven by Stripe, make sure that you activated appropriate module. Go to **Advanced Settings** page and turn on



[Activate Stripe Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Advanced](#) tab, click [Payment Gateways](#) and click [+ Stripe](#) button. New gateway based on Stripe will be added. Click  to configure it.

*Important! Make sure that you created webhook with the URL provided by plugin for event `checkout.session.completed` in your Stripe Dashboard.*

It has the following parameters:

- **Name.** The name of the payment gateway. It is used for your convenience.
- **Publishable Key.** Enter valid Publishable Key. Find it on API Keys page in Stripe Dashboard.
- **Secret Key.** Enter valid Secret Key. Find it on API Keys page in Stripe Dashboard.
- **Signing secret.** Enter a valid Signing secret for webhook that you created earlier. Find it on Webhooks page in Stripe Dashboard. Click webhook that you created earlier, and find "Signing secret" parameter.
- **Item name.** Enter the item name.
- **Amount.** Set amount to pay and currency.
- **Successful payment URL.** All payers are redirected to this URL after successful payment.
- **Cancel URL.** All payers are redirected to this URL in case of failed/cancelled payment.

## Yandex.Money

Before creating payment gateway driven by Yandex.Money, make sure that you activated appropriate module. Go to [Advanced Settings](#) page and turn on [Activate Yandex.Money Integration module](#). Then go to form editor, click  button on Top Toolbar, select [Advanced](#) tab, click [Payment Gateways](#) and click [+ Yandex.Money](#) button. New gateway based on Yandex.Money will be added. Click  to configure it.

*Important! Make sure that you set URL provided by plugin as URL for HTTP-notices in your Yandex.Money Dashboard.*

It has the following parameters:

- **Name.** The name of the payment gateway. It is used for your convenience.
- **Wallet ID.** Enter your Wallet ID (number). Find it on Settings page in Yandex.Money Account.
- **Secret Word.** Enter valid Secret Word. The Secret Word allows you to verify the authenticity of the notifications. Find it on the same page where you entered URL for HTTP-notices in your Yandex.Money Dashboard.
- **Label prefix.** Enter a random alphanumeric string. It is required for some intermediate services to split HTTP-notices. If you do not use them, just ignore

this parameter and leave it as is.

- **Payment type.** select the payment type (Wallet or Bank Card).
- **Item name.** Enter the item name.
- **Amount.** Set amount to pay and currency.
- **Successful payment URL.** All payers are redirected to this URL after successful payment.


## 4.10 CUSTOM JAVASCRIPT HANDLERS

---

You can use your own JavaScript-code for some form events. It doesn't mean that native JavaScript handlers are replaced by yours. It means that your JavaScript-code is executed together with native handlers.

To enable Custom JavaScript Handlers, make sure that you activated appropriate module. Go to [Advanced Settings](#) page and turn on

[Activate Custom JavaScript Handlers module](#)

. Then go to form editor, click  button on Top Toolbar, select [Advanced](#) tab and click

 [Custom JavaScript Handlers](#)

. There you can enable certain handler and put JS-code.

At that moment you can add JavaScript-code to the following events:

- **AfterInit.** This handler is executed when form initialized.
- **AfterSubmitSuccess.** This handler is executed when form successfully submitted.

The following variables and functions might be useful in your JavaScript-code:

- `this.dom_id` - this ID is used to access form DOM elements.
- `this.form_id` - form ID.
- `this.user_data` - object, that can be used to store your data.
- `this.get_field_value(ELEMENT_ID)` - get the value of certain input field.

Replace `ELEMENT_ID` by real field ID taken from Advanced tab on field properties window.

- `this.set_field_value(ELEMENT_ID, value)` - set the value of certain input field. Replace `ELEMENT_ID` by real field ID taken from Advanced tab on field properties window.

*Do not use `<script>...</script>` tags inside (just put regular JavaScript-code) and make sure your JavaScript-code doesn't have any syntax errors.*



















## 5 FORM ELEMENTS

### 5.1 GENERAL INFO

---

Each form consists of Form Elements. You can add as many Form Elements as you need and group them into complex layouts including unlimited column nesting. If you want to add some element to the form (for example, input field or button or something else), just click relevant icon on Form Elements Toolbar and new element will be added. Then you can move new element into desired position and configure it.


Following elements can be added to the form:

-  - Text Input Field
-  - Email Input Field
-  - Textarea
-  - Select Box (Drop-down List)
-  - Checkbox
-  - Radio Button
-  - Multiselect List
-  - Image Select
-  - Date Picker
-  - Time Picker
-  - File Upload
-  - Password
-  - Star Rating
-  - Hidden Field
-  - Submit Button
-  - Column Layouts
-  - HTML Content
-  - Link Button

The properties of each element are described below.

### 5.2 TEXT FIELD

---

Click  on Form Elements Toolbar to add regular text field to the form. To achieve its properties just do right click over element and then click [Properties](#). All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

## Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Placeholder.** The placeholder text will appear inside the field until the user starts to type.
- **Autocomplete attribute.** Choose the value of the autocomplete attribute. It helps browser to fill the field value, if required.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

## Style Properties

The general view of text fields is configured through [Form Settings](#). It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Input style.** Adjust the input field style (size and text alignment).
- **Input icons.** These icons appear inside/near of the input field.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **Custom CSS class.** This class name will be added to the input field.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- **Mask.** Adjust the mask of the input field. Use the following special symbols:

0 - mandatory digit

9 - optional digit

A - alphanumeric character

S - alpha character

*Important! jQuery Mask plugin must be enabled on [General Settings](#) page.*

- **Default value.** The default value is the value that the field has before the user has entered anything.

- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Read only.** If enabled, the user can not edit the field value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.
- **Filters.** Filters allow you to strip various characters from the submitted value.



The following filters are available:

- **Alpha.** Removes any non-alphabet characters.
- **Alphanumeric.** Removes any non-alphabet characters and non-digits.
- **Digits.** Removes non-digits.
- **Regex.** Removes characters matching the given regular expression.
- **Strip Tags.** Removes any HTML tags.
- **Trim.** Removes white space from the start and end.
- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:
  - **Alpha.** Checks that the value contains only alphabet characters.
  - **Alphanumeric.** Checks that the value contains only alphabet characters or digits.
  - **Digits.** Checks that the value contains only digits.
  - **Email.** Checks that the value is a valid email address.
  - **Equal.** Checks that the value is identical to the given token.
  - **Greater Than.** Checks that the value is numerically greater than the given minimum.
  - **In Array.** Checks that the value is in a list of allowed values.
  - **Length.** Checks that the length of the value is between the given maximum and minimum.
  - **Less Than.** Checks that the value is numerically less than the given maximum.
  - **Prevent Duplicates.** Checks that the same value has not already been submitted.

- **Regex.** Checks that the value matches the given regular expression.
- **URL.** Checks that the value is a valid URL.

## 5.3 EMAIL FIELD


---

Click  on Form Elements Toolbar to add email field to the form. To achieve its properties just do right click over element and then click . All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

### Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Placeholder.** The placeholder text will appear inside the field until the user starts to type.
- **Autocomplete attribute.** Choose the value of the autocomplete attribute. It helps browser to fill the field value, if required.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

### Style Properties

The general view of text fields is configured through . It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Input style.** Adjust the input field style (size and text alignment).
- **Input icons.** These icons appear inside/near of the input field.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **Custom CSS class.** This class name will be added to the input field.
- **CSS styles.** Set custom css for certain element parts and states.

### Data Properties

- **Default value.** The default value is the value that the field has before the user has entered anything.



- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Read only.** If enabled, the user can not edit the field value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.


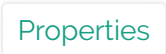
## Advanced Properties

- **ID.** The unique ID of the input field.
- **Filters.** Filters allow you to strip various characters from the submitted value.

The following filters are available:

- **Alpha.** Removes any non-alphabet characters.
- **Alphanumeric.** Removes any non-alphabet characters and non-digits.
- **Digits.** Removes non-digits.
- **Regex.** Removes characters matching the given regular expression.
- **Strip Tags.** Removes any HTML tags.
- **Trim.** Removes white space from the start and end.
- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:
  - **Email.** Checks that the value is a valid email address.
  - **Equal.** Checks that the value is identical to the given token.
  - **In Array.** Checks that the value is in a list of allowed values.
  - **Prevent Duplicates.** Checks that the same value has not already been submitted.

## 5.4 TEXTAREA

Click  on Form Elements Toolbar to add textarea to the form. To achieve its properties just do right click over element and then click . All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

## Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Placeholder.** The placeholder text will appear inside the field until the user starts to type.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

## Style Properties

The general view of text fields is configured through [Form Settings](#). It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Textarea style.** Adjust the input field style (size and text alignment).
- **Textarea icons.** These icons appear inside/near of textarea.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **Custom CSS class.** This class name will be added to the input field.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- **Default value.** The default value is the value that the field has before the user has entered anything.
- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Read only.** If enabled, the user can not edit the field value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.

- **Filters.** Filters allow you to strip various characters from the submitted value.

The following filters are available:



- **Alpha.** Removes any non-alphabet characters.
- **Alphanumeric.** Removes any non-alphabet characters and non-digits.
- **Digits.** Removes non-digits.
- **Regex.** Removes characters matching the given regular expression.
- **Strip Tags.** Removes any HTML tags.
- **Trim.** Removes white space from the start and end.

- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:

- **Alpha.** Checks that the value contains only alphabet characters.
- **Alphanumeric.** Checks that the value contains only alphabet characters or digits.
- **Digits.** Checks that the value contains only digits.
- **Email.** Checks that the value is a valid email address.
- **Equal.** Checks that the value is identical to the given token.
- **Greater Than.** Checks that the value is numerically greater than the given minimum.
- **In Array.** Checks that the value is in a list of allowed values.
- **Length.** Checks that the length of the value is between the given maximum and minimum.
- **Less Than.** Checks that the value is numerically less than the given maximum.
- **Prevent Duplicates.** Checks that the same value has not already been submitted.
- **Regex.** Checks that the value matches the given regular expression.
- **URL.** Checks that the value is a valid URL.

## 5.5 SELECT BOX (DROP-DOWN LIST)

---

Click  on Form Elements Toolbar to add select box (drop-down list) to the form. To achieve its properties just do right click over element and then click . All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

### Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Options.** These are the choices that the user will be able to choose from. You can add as many options as you need manually or insert a set of pre-defined options such as countries, states, provinces, etc.
- **"Please select" option.** Adds an option to the top of the list to let the user choose no value.
- **Autocomplete attribute.** Choose the value of the autocomplete attribute. It helps browser to fill the field value, if required.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

## Style Properties

The general view of input fields is configured through [Form Settings](#). It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Input style.** Adjust the input field style (size and text alignment).
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **Custom CSS class.** This class name will be added to the input field.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Submit on choice.** If enabled, the form will be submitted when an option is chosen.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic



Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.
- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:
  - **Equal.** Checks that the value is identical to the given token.
  - **Greater Than.** Checks that the value is numerically greater than the given minimum.
  - **In Array.** Checks that the value is in a list of allowed values.
  - **Length.** Checks that the length of the value is between the given maximum and minimum.
  - **Less Than.** Checks that the value is numerically less than the given maximum.
  - **Prevent Duplicates.** Checks that the same value has not already been submitted.
  - **Regex.** Checks that the value matches the given regular expression.

## 5.6 CHECKBOX


---

Click  on Form Elements Toolbar to add a set of checkboxes to the form. To achieve its properties just do right click over element and then click . All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

### Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Options.** These are the choices that the user will be able to choose from. You can add as many options as you need manually or/and insert a set of pre-defined options such as countries, states, provinces, etc.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

### Style Properties

The general view of checkboxes is configured through . It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Checkbox style.** Choose how to display checkbox fields and their captions.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic


Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.
- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:
  - **In Array.** Checks that the value is in a list of allowed values.
  - **Prevent Duplicates.** Checks that the same value has not already been submitted.

## 5.7 RADIO BUTTON

---

Click  on Form Elements Toolbar to add a set of radio buttons to the form. To achieve its properties just do right click over element and then click [Properties](#). All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

### Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Options.** These are the choices that the user will be able to choose from. You can add as many options as you need manually or/and insert a set of pre-defined options such as countries, states, provinces, etc.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

## Style Properties

The general view of radio buttons is configured through [Form Settings](#). It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Radio button style.** Choose how to display radio button fields and their captions.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.


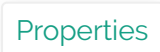
## Advanced Properties

- **ID.** The unique ID of the input field.
- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:
  - **In Array.** Checks that the value is in a list of allowed values.

- **Prevent Duplicates.** Checks that the same value has not already been submitted.

## 5.8 MULTISELECT


---

Click  on Form Elements Toolbar to add a multiselect field to the form. To achieve its properties just do right click over element and then click . All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

### Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Options.** These are the choices that the user will be able to choose from. You can add as many options as you need manually or/and insert a set of pre-defined options such as countries, states, provinces, etc.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

### Style Properties

The general view of input fields is configured through . It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Multiselect style.** Choose how to display multiselect options.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **CSS styles.** Set custom css for certain element parts and states.

### Data Properties

- **Maximum selected options.** Enter how many options can be selected. Set 0 for unlimited number.
- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.



- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic



Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.
- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:
  - **In Array.** Checks that the value is in a list of allowed values.
  - **Prevent Duplicates.** Checks that the same value has not already been submitted.

## 5.9 IMAGE SELECT

---

Click  on Form Elements Toolbar to add a set of images to the form. To achieve its properties just do right click over element and then click . All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

### Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Mode.** Image Select can operate in 2 modes: radio button (user can select only one option) and checkbox (user can select multiple options).
- **Options.** These are the choices that the user will be able to choose from. You can add as many options as you need.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

### Style Properties

The general view of the field is configured through [Form Settings](#). It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Image style.** Choose how to display images.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic


Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.
- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:
  - **In Array.** Checks that the value is in a list of allowed values.
  - **Prevent Duplicates.** Checks that the same value has not already been submitted.

### 5.10 DATE PICKER

---

Click  on Form Elements Toolbar to add date picker to the form. To achieve its properties just do right click over element and then click [Properties](#). All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

## Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Placeholder.** The placeholder text will appear inside the field until the user starts to type.
- **Autocomplete attribute.** Choose the value of the autocomplete attribute. It helps browser to fill the field value, if required.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

## Style Properties

The general view of input fields is configured through [Form Settings](#). It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Input style.** Adjust the input field style (size and text alignment).
- **Input icons.** These icons appear inside/near of the input field.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **Custom CSS class.** This class name will be added to the input field.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- **Default value.** The default value is the value that the field has before the user has entered anything.
- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Minimum date.** Adjust the minimum date that can be selected.
- **Maximum date.** Adjust the maximum date that can be selected.
- **Read only.** If enabled, the user can not edit the field value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties


- **ID.** The unique ID of the input field.
- **Filters.** Filters allow you to strip various characters from the submitted value.

The following filters are available:

- **Alpha.** Removes any non-alphabet characters.
  - **Alphanumeric.** Removes any non-alphabet characters and non-digits.
  - **Digits.** Removes non-digits.
  - **Regex.** Removes characters matching the given regular expression.
  - **Strip Tags.** Removes any HTML tags.
  - **Trim.** Removes white space from the start and end.
- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:
    - **Date.** Checks that the value is a valid date (according to pre-defined date format set on [Form Settings](#) ).

### 5.11 TIME PICKER

---

Click  on Form Elements Toolbar to add time picker to the form. To achieve its properties just do right click over element and then click [Properties](#) . All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

#### Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Placeholder.** The placeholder text will appear inside the field until the user starts to type.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

#### Style Properties

The general view of input fields is configured through [Form Settings](#) . It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Input style.** Adjust the input field style (size and text alignment).
- **Input icons.** These icons appear inside/near of the input field.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **Custom CSS class.** This class name will be added to the input field.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- **Default value.** The default value is the value that the field has before the user has entered anything.
- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Minimum time.** Adjust the minimum time that can be selected.
- **Maximum date.** Adjust the maximum time that can be selected.
- **Minute interval.** Enter the minute interval.
- **Read only.** If enabled, the user can not edit the field value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.
- **Filters.** Filters allow you to strip various characters from the submitted value.


The following filters are available:

- **Alpha.** Removes any non-alphabet characters.
- **Alphanumeric.** Removes any non-alphabet characters and non-digits.
- **Digits.** Removes non-digits.
- **Regex.** Removes characters matching the given regular expression.
- **Strip Tags.** Removes any HTML tags.
- **Trim.** Removes white space from the start and end.
- **Validators.** Validators check whether the data entered by the user is valid. If not, then appropriate error bubble is displayed for this field. The following validators are available:

- **Time.** Checks that the value is a valid time (according to pre-defined time format set on [Form Settings](#) ).

## 5.12 FILE UPLOAD

---

Click  on Form Elements Toolbar to add file upload field to the form. To achieve its properties just do right click over element and then click [Properties](#) . All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

### Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Caption.** This is the caption of upload button.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

### Style Properties

The general view of button is configured through [Form Settings](#) . It's applied to all buttons. Here you can adjust view of the button.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Button style.** Adjust the button style and position.
- **Icons.** These icons appear near the button caption.
- **Colors.** Adjust the colors of the button. You can do it for different button states.
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **Custom CSS class.** This class name will be added to the input field.
- **CSS styles.** Set custom css for certain element parts and states.

### Data Properties

- **Allowed extensions.** Enter the comma-separated list of allowed file extensions. Due to security reasons, never allow executable extensions such as .php, .js, .exe, .msi, etc. You also can adjust the text of error message that appears if user tries to upload not allowed files.

- **Maximum allowed size** Enter the maximum size of a file in MB. Don't set it higher than the value of `upload_max_filesize` in your PHP settings. You also can adjust the text of error message that appears if user tries to upload big file.
- **Maximum number of files.** Enter the maximum number of files that can be uploaded by user. You also can adjust the text of error message that appears if user tries to upload more files then maximum number of files.

## Conditional Logic


Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.

### 5.13 PASSWORD

---

Click  on Form Elements Toolbar to add password field to the form. To achieve its properties just do right click over element and then click `Properties`. All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

## Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.
- **Label.** This is the label of the field.
- **Placeholder.** The placeholder text will appear inside the field until the user starts to type.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

## Style Properties

The general view of input fields is configured through `Form Settings`. It's applied to all input fields. Here you can adjust view of the field.

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Input style.** Adjust the input field style (size and text alignment).
- **Input icons.** These icons appear inside/near of the input field.

- **Description style.** Choose where to display the description relative to the field and its alignment.
- **Custom CSS class.** This class name will be added to the input field.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- Enter the minimum password length. You also can adjust the text of error message that appears if submitted password is too short.
- **Capital letters is mandatory.** If enabled, the password must contains at least one capital letter. You also can adjust the text of error message that appears if submitted password doesn't contain capital letter.
- **Digit is mandatory.** If enabled, the password must contains at least one digit. You also can adjust the text of error message that appears if submitted password doesn't contain digit.
- **Special character is mandatory.** If enabled, the password must contains at least one special character: !\$#%^&\*~\_~(){}[]\|/? . You also can adjust the text of error message that appears if submitted password doesn't contain special character.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic


Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.

### 5.14 STAR RATING

---

Click  on Form Elements Toolbar to add star rating field to the form. To achieve its properties just do right click over element and then click [Properties](#) . All properties are organized into 5 groups: Basic, Style, Data, Logic and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

## Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.



- **Label.** This is the label of the field.
- **Description.** This description appears below the field.
- **Required.** If enabled, the user must fill out the field. You also can specify error message here.

## Style Properties

- **Label style.** Choose where to display the label relative to the field and its alignment.
- **Star style.** Adjust the style of stars (size, alignment, colors).
- **Description style.** Choose where to display the description relative to the field and its alignment.
- **CSS styles.** Set custom css for certain element parts and states.

## Data Properties

- **Number of stars.** Choose the total number of stars.
- **Default value.** The default value is the value that the field has before the user has entered anything.
- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Conditional Logic


Here you can create rules to show or hide the element depending on the values of other fields.

## Advanced Properties

- **ID.** The unique ID of the input field.

### 5.15 HIDDEN FIELD

---

Click  on Form Elements Toolbar to add hidden field to the form. To achieve its properties just do right click over element and then click **Properties**. All properties are organized into 3 groups: Basic, Data and Advanced. To view/edit certain group of properties just click appropriate tab on Element Properties window.

## Basic Properties

- **Name.** The name will be shown in place of the label throughout the plugin, in the notification email and when viewing submitted form entries.

## Data Properties


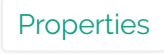
- **Default value.** The default value is the value that the field has before the user has entered anything.
- **Dynamic default value.** Allows the default value of the field to be set dynamically via a URL parameter.
- **Parameter name.** This is the name of the parameter that you will use to set the default value.
- **Save to database.** If enabled, the submitted element data will be saved to the database and shown when viewing an entry.

## Advanced Properties

- **ID.** The unique ID of the input field.

### 5.16 BUTTON


---

Click  on Form Elements Toolbar to add submit button to the form. To achieve its properties just do right click over element and then click . All properties are organized into 3 groups: Basic, Style and Logic. To view/edit certain group of properties just click appropriate tab on Element Properties window.

## Basic Properties

- **Name.** The name is used for your reference.
- **Label.** This is the label of the button.
- **Type.** Choose the type of the button (back, next or submit).
- **Sending label.** This is the label of the button when data are sending to server.

## Style Properties

The general view of button is configured through . It's applied to all buttons. Here you can adjust view of the field.


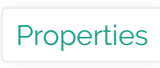
- **Style.** Adjust the button style and position.
- **Icons.** These icons appear near the button label.
- **Colors.** Adjust the colors of the button. You can do it for different button states.
- **Custom CSS class.** This class name will be added to the button.
- **CSS styles.** Set custom css for certain element parts and states.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.

### 5.17 CUSTOM HTML

---

Click  on Form Elements Toolbar to custom HTML code to the form. To achieve its properties just do right click over element and then click . All properties are organized into 3 groups: Basic, Style, Logic. To view/edit certain group of properties just click appropriate tab on Element Properties window.

#### Basic Properties

- **Name.** The name is for your reference only.
- **HTML.** This is the HTML-content of the element.

#### Style Properties


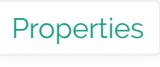
- **CSS styles.** Set custom css for certain element parts.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.

### 5.18 LINK BUTTON

---

Click  on Form Elements Toolbar to add link button to the form. To achieve its properties just do right click over element and then click . All properties are organized into 3 groups: Basic, Style and Logic. To view/edit certain group of properties just click appropriate tab on Element Properties window.

#### Basic Properties

- **Name.** The name is used for your reference.
- **Label.** This is the label of the button.
- **URL.** Specify the URL where users redirected to.
- **Open link in new tab.** If enabled, the link will be opened in new tab.

#### Style Properties

The general view of button is configured through [Form Settings](#). It's applied to all buttons. Here you can adjust view of the button.


- **Style.** Adjust the button style and position.
- **Icons.** These icons appear near the button label.
- **Colors.** Adjust the colors of the button. You can do it for different button states.
- **Custom CSS class.** This class name will be added to the button.
- **CSS styles.** Set custom css for certain element parts and states.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.

### 5.19 COLUMN LAYOUTS

---

Hover mouse over  on Form Elements Toolbar, and select how many columns you want to add to the form. Once clicked, new column layout will added. You can move any form elements (including other column layouts) into columns. To achieve its properties just do right click over element and then click [Properties](#). All properties are organized into 2 groups: Basic and Logic. To view/edit certain group of properties just click appropriate tab on Element Properties window.

## Basic Properties

- **Name.** The name is for your reference only.
- **Column width.** Specify the width of each column. The row is divided into 12 equal pieces. You can decide how many pieces related to each columns. If you want all columns to be in one row, make sure that sum of widths is equal to 12.

## Conditional Logic

Here you can create rules to show or hide the element depending on the values of other fields.