



Simpleview CRM Web Form API v. 1.0

GETTING STARTED

Dear Developer,

Thank you for your interest in the Simpleview CRM API for Web Forms. This version of the API is currently limited to pulling pre-built forms created within CRM and then posting the information directly back to CRM.

The following steps are required in order to get started. Have your client contact their project manager to set up your API Login and password and provide you with the client's base API URL. Once that is complete and you have this document in hand, you are ready to go. The base URL will differ from client to client, but assuming Simpleview was the client, the URL would likely look like **<https://api.simpleviewinc.com>**.

If you have already begun using our Membership Management API, then these initial steps should have already been completed and you will use your same username and password to access this API.

At the moment there are no code samples available for interfacing with the API, but we hope to get some set up shortly in most of the common web development languages such as PHP, JSP, ASP.NET, and Coldfusion.

UPDATES TO THE API AND THIS DOCUMENT

This version of the API as well as its documentation may be updated from time to time. This would encompass handling various tasks such as bug fixes and omissions. We will make every effort to alert and give you advanced warning before any changes are being made that may affect your implementation. This is likely to be limited to adding additional data or at worst minor modifications to formatting.

WHAT TO SEND AND WHERE TO SEND IT

WHERE TO SEND IT

All requests sent for form data should be sent to the following URL:

<<clientbaseurl>>/webapi/forms

WHAT TO SEND

The method invoked is specified by the file you call in the URL (ex. getform.cfm). You can view more detail on the methods available later in this document. All values should be sent via HTTP POST parameters. Any XML values should be sent with the full XML header and should always be encoded in UTF-8. Additional requirements may be specified in the future.

All requests require the following parameters:

username: <i>String</i>	The username you were assigned
password: <i>String</i>	The password you were assigned

GENERAL RESPONSE INFORMATION

For the form API, the response returned contains the complete HTML of the requested form. The API will return all necessary JavaScript validation and employs CSS to format the layout and formatting of the form. More details on the CSS class names available in Appendix A.

METHOD OVERVIEW

The following methods are supported and documented in this API

- GetForm

API METHODS

GETFORM – RETURNS HTML

FILENAME: GETFORM.CFM

URL:

<<clientbaseurl>>/webapi/forms/getform.cfm

PARAMETERS:

STRING USERNAME

STRING PASSWORD

INTEGER FORMID

DESCRIPTION:

This is the method used to return the web form based on the form ID passed to the call. This will return the complete HTML, JavaScript, etc. to have a fully functioning form. The form will have the proper action to send the submission back to the form API, which will in turn validate it and insert the data into CRM.

RESPONSE:

If the username and password are correct AND the form ID passed relates to a valid and active web form, then the response will return the raw HTML. If there is any error related to retrieving the form, the response will return a status code of **500** in the header and the content will consist of:

[Error Code] XXX

Details on each error code are listed under Appendix B.

FORM SUBMISSION:

The <form> tag returned in the response HTML code will be pre-set with the proper “action” to submit the form back to the API. Once a form is submitted, assuming it passes all JavaScript validation, the POST data is automatically sent back to the API URL to insert the data into CRM and send a notification email (optional).

There are 2 settings available to every web form that determines what to do after the form has been submitted. These settings are setup upon the form creation in the Form Builder module (See below) and can be different for each web form that is created.

SUCCESS URL:

This setting contains the URL the user should be redirected to after a successful form submission. This should be a URL on the client website and can be used to display a thank you page, or it can be used to redirect the user to another section of the website.

ERROR URL:

This setting contains the URL the user should be redirected to if there is a server-side validation error in the form submission. The API will append an error code to the URL using “errcode” as the URL parameter. For example, a sample URL would be: <http://www.domain.com/?errcode=XXX>.

NOTE: If an Error URL is not setup, the error code will be returned to the browser.

FORM BUILDER API MODULE

Within Simpleview CRM there is a module available to create custom forms based on the available tables and fields within CRM. We have also added access this module within the web form API. You can access this module from here:

<<clientbaseurl>>/webapi/forms/admin/

This URL will prompt you for your API username and password, and once logged in you will be able to view all the pre-existing web forms setup within CRM, and have the ability to add or modify any form. Each form consists of general settings, one or more form sections, form questions that will appear on the form, and form defaults.

GENERAL SETTINGS:

For each form you can give it a title for reference, set a date range of when the form should be active, add one or more email addresses to receive a notification email upon each form submission, and set your Error URL and Success URL.

FORM SECTIONS:

You need to create at least one section, as form questions are contained under form sections. The section controls the layout (1-column, 2-column, etc.), displays a section name and can also show hint(s) or notes above the field questions contained with the section.

FORM QUESTIONS:

Form questions are selected from available fields (based on the form type) and assigned to a form section. Each question has many options available, such as: display name, hint, sort order, required status, default values, layout, JavaScript options, etc.

FORM DEFAULTS:

Form defaults are fields that will not appear on the form, but need to contain a value when the data is inserted into the CRM database. If a required field is not shown on the form as a question, then it must be a default and have a value assigned.

APPENDIX A: CSS CLASSES

Below is a list of CSS classes available to customize the look/layout of the form on the client website:

Class Name	Description
.sv_api_form	Assigned to <form> tag
.sv_api_main_tbl	Assigned to <table> tag for main table containing entire form
.sv_api_section_tbl	Assigned to <table> tag for each form section
.sv_api_checkbox_tbl	Assigned to <table> tag for display of multi-column check boxes (for multi-select fields only)
.sv_api_reqhint	Assigned to the <td> containing the hint at the bottom of the form informing the user of the required fields
.sv_api_submit	Assigned to the <td> containing the form buttons (Submit, Reset)
.sv_api_section_head	Assigned to the <td> containing the section title
.sv_api_section_body	Assigned to the <td> containing the section body (encompassing the section table)
.sv_api_section_note	Assigned to the <td> containing the section hint/notes
.sv_api_field_disp	Assigned to the <td> containing each question label
.sv_api_field_val	Assigned to the <td> containing each question field (ex. input, select, etc.)
.sv_api_field_required	Assigned to the <td> containing required fields (this is added in addition to .sv_api_field_disp for required fields only)
.sv_api_button	Assigned to <input type="Submit/Reset"> tags
.sv_api_checkbox	Assigned to <input type="Checkbox"> tags
.sv_api_input	Assigned to <input type="Text"> tags
.sv_api_label	Assigned to <label> tags
.sv_api_radio	Assigned to <input type="Radio"> tags
.sv_api_select	Assigned to <select> tags
.sv_api_textarea	Assigned to <textarea> tags
.sv_api_hint	Assigned to any tag containing a field hint

None of these classes are required, and you should only create the classes you need in your site's CSS. We have also provided a sample CSS file to layout the form using a default style. This sample CSS can be found at:

<<clientbaseurl>>/webapi/forms/includes/default.css

APPENDIX B: ERROR CODES

Below is a list of error codes that may be returned:

Error Code	Description
100	Form ID was not passed to method or is not found in database
101	Form is either expired or has not begun, based on date range setup within Form Builder.
102	Form was found, but is not a valid web form
200	Username and/or password were not passed to method
201	Username/Password authentication failed
300	Form submission failed Captcha verification
500	Unknown or Unhandled Error

APPENDIX C: CODE SAMPLES

A, PHP Using CURL

```
define('POSTURL', '<<clientbaseurl>>/webapi/forms/getform.cfm');
define('POSTVARS', 'username=<<clientusername>> &password<<clientpassword>>/FORMID=<<formid>>'); // POST VARIABLES TO BE SENT

$ch = curl_init(POSTURL);
curl_setopt($ch, CURLOPT_POST,1); //THE REQUEST MUST USE POST
curl_setopt($ch, CURLOPT_POSTFIELDS ,POSTVARS); //ARGUMENTS SENT VIA POST
curl_setopt($ch, CURLOPT_FOLLOWLOCATION ,1);
curl_setopt($ch, CURLOPT_HEADER ,0); // DO NOT RETURN HTTP HEADERS
curl_setopt($ch, CURLOPT_RETURNTRANSFER ,1); // RETURN THE CONTENTS OF THE CALL
$Rec_Data = curl_exec($ch);
echo $Rec_Data; //THIS WILL BE YOUR FORM VARIABLE
curl_close($ch);
```