

Generic Form Handler

The Generic Form Handler (GFH) is software hosted on iPost's servers. Used in conjunction with the many robust features contained in iPost, it allow you to create sophisticated marketing programs quickly and easily.

The GFH will process data gathered on a web hosted form and insert that data into a list or data table. Its features include:

- Saving Customer Data
- Maintaining Customer Preferences
- Trigger Journey Automations
- Building Mailing Segments
- Retrieving Customer Data Using an Automated System

! Warning: Use of the Generic Form Handler via 'front-side', or 'browser-served' javascript can create a condition where a third-party can manipulate form data. iPost recommends that your form data be processed from a non-web accessible area of your host. For Best Practices regarding the implementation of a javascript manipulated form to mitigate the XCRF vulnerability, please read [here](#) for more information

How does it work?

At its most basic level, each form on the web requires at least two parts:

1. The user interface, or HTML form, with which the user interacts
2. The software which knows what to do with the data collected on the HTML form. This is the form handler.

Every HTML form most contain an *action=* attribute on the *<FORM>* tag. This attribute tells the web server the name and location of the form handler. Through the HTML form and its handler often are hosted on the same server, this is not required.

To use iPost's GFH a web designer simply sets the value of the *action=* attribute to the URL for the handler and encodes the form inputs using the correct names.

Enabling the Generic Form Handler

To enable this feature, follow these steps.


1. Go to the *Settings* menu
2. Select the *Database* option

3. Check the *Enable Use of the Generic Form Handler* box
4. Check the box *Enforce GFH Key match on access* if retrieving data from iPost with the GFH
5. Enter the domains that the GFH will be retrieving or posting data form in the *Limit GFH redirection to these domains* box.
6. Click *Save*

Using the Generic Form Handler to Save Data

Se the `action=` attribute of your `<FORM>` tag to one of the URLs below depending on your iPost instance:


```
https://g001.enterprise.ipost.com/forms.cgi
```


 Click to copy

```
https://g002.enterprise.ipost.com/forms.cgi
```

 Click to copy

```
https://g003.enterprise.ipost.com/forms.cgi
```

 Click to copy

 Please reach out to support@ipost.com if you are unsure of which your instance your account is on.

For security reasons, we recommend that you use the POST method for submitting data to the GFH. Below is an example of a correctly formatted `<FORM>` tag:

```
<FORM method="POST" id="example_form" action="https://g001.enterprise.ipost.com/forms.cgi">
```

 Click to copy

If your form will be linked from a mailing document used in iPost, you may also wish to refer to Encrypted Contact Identity.

Required Parameters

Each form which use the GFH **must** contain the following inputs.

Input	Definition
client_token	The <i>client_token</i> is a unique abbreviation used to represent your client in the system. It is chosen by you or by staff when your client account

Input	Definition
	is set up. You can see your client token by clicking on the Client tab under Settings, where Client Token is displayed near the top.
success_url	This is the URL for the webpage to which the GFH will redirect the user following a successful submission of the form.
error_url	This is the URL for the webpage to which the Generic Form Handler will send the user if it encounters any errors while processing the data on the form. The error conditions and how to display the error messages on this webpage are discussed in detail below.
email_addr	The email address of the user submitting the form.
submit or btnSubmit	The submission mechanism for the form.
subscribe or listName	In order to create a new contact (i.e., for a previously-unrecorded email address), one of <i>subscribe</i> or <i>listName</i> is required.

Optional Parameters

The GFH supports many optional parameters, enabling a wealth of valuable features. The list of optional parameters are listed below.

Email List Data Fields

The GFH can save data to most of the Email List Data fields which have been set up in [Database Settings](#) page in your account.

The name of the input on the form will be the same as the iTL Macro for the associated field on the Database page. For example, a First Name field may have an iTL Macro of first_name. The form input for the same field would look something like this:

```
<INPUT type="text" name="first_name" size="40" maxlength="255">
```

 Click to copy

Email List Topics

The GFH can store subscriber newsletter preferences in the form of Topics. The name of the input in the form will be topic_ immediately followed by the ID number of the associated topic. Topical inputs **MUST** have a value of either 1 or 0. A value of 1 means that the user is interested in that topic.

For example, a topic named "Camping/Hiking" with an ID number of 1 may have a form input something like this:

```
<INPUT type="checkbox" name="topic_1" value="1" />
```

 Click to copy

Email Subscription Preference

The GFH can be used to maintain your users' email subscription preferences.

The input name for subscription preferences must be subscribe. It **must** have one of the following three values:

1. optin: opts the user IN to receive emails
2. optout: opts the user OUT of receiving emails
3. change: allows the user to remain opted in but transfer the subscription to a new email address.

If the value of change is used then the form **MUST** also include an input named new_addr. This is the address to which the user wishes to transfer their subscription.

PLEASE NOTE: A subscribe input which contains a value other than the three above will be interpreted as an opt-out.

Example inputs which incorporate all of these subscription options:

```
<INPUT type="radio" name="subscribe" value="optin">&nbsp;Yes, please send me  
information!<br />  
<INPUT type="radio" name="subscribe" value="optout">&nbsp;No, thanks.<br />  
<INPUT type="radio" name="subscribe" value="change">&nbsp;I already receive emails from  
Zanshin Outfitters but I need to change my address to:<br />  
<INPUT type="text" name="new_addr" size="40" maxlength="255" />
```

 Click to copy

The subscribe input will automatically add the contact to your organization's *All Contacts* email list.

Email Format


iPost supports sending messages in HTML, text, and multipart formats. Though the default is HTML, some users may prefer to receive messages in text-only format instead.

If you would like to allow the user to select an email format preference, include an input named email_fmt in the form. This input can have two possible values:

1. html
2. text

For example:

```
<INPUT type="radio" name="email_fmt" value="html">&nbsp;HTML<br />
<INPUT type="radio" name="email_fmt" value="text">&nbsp;Text-only<br />
```

 Click to copy


Email List Name

The GFH can add a subscriber to, or remove a subscriber from, a list of your choice.

The name of the input must be `listName`. The value of the input must be the name of the list **exactly** as it appears in the List Manager.

For example


```
<input type="hidden" name="listName" value="Spring Contest Entries" />
```

 Click to copy

To add a subscriber to more than one list, simply add multiple input fields. All should be named `listName` and values containing **exact** list names.

For example:

```
<input type="hidden" name="listName" value="Spring Contest Entries" />
<input type="hidden" name="listName" value="Regional Promotions" />
```

 Click to copy

To remove a subscriber from one or more lists, add an input named `remove` with the value 1. This input will be ignored if a `listName` is not also present. *Note that it is **not** possible to both add a subscriber to one list and remove him from another list in a single form submission.*

Journey Automation Initiation

The GFH can submit an address for Journey Automation upon successful form submission.

To initiate a contact upon a Journey via the GFH create an active "API On Demand" Journey Automation in Automation and note the Journey ID via the URL. Supply this integer value to the GFH via the `journey_id` parameter.

For example:

```
<INPUT type="hidden" name="journey_id" value="99999999" />
```

 Click to copy

Debugging

If the form contains an input with a name of debug with a value of 1 then the data contained on the form will NOT be saved upon form submission and no mailings will be triggered. Instead the form's parameters will be returned to the browser as a simple HTML page containing parameter = value pairs.

For example:

```
<INPUT type="hidden" name="debug" value="1" />
```

 Click to copy

A value of 2 returns additional information about the gfh_key parameter.

Disabling Error URL Rewriting

In order to display the ipost_error_detail DIV it is necessary for the Generic Form Handler to rewrite the HTML source of your error page to insert the new element. This may not work for some webpages, causing them to display improperly. If this is the case for your error page, you may place an input named redirect_gfh_errors in your form. You must set a value for this input, but any non-empty value is accepted. If this input is on the form then the Generic Form Handler performs a simple redirect to the error page rather than rewriting it. The disadvantage of this method is that there is **NO WAY** to view the form errors without rewriting the page. Please keep this in mind before including redirect_gfh_errors in the form.

An example:

```
<INPUT type="hidden" name="redirect_gfh_errors" value="yes" />
```

 Click to copy

Disabling Creation of New Subscribers

While typically you will wish your form to create a new subscriber in the case than an email address is entered of which has no record, there may be a case where you will wish the form to notify you in this situation. You may place an input named ncreate in your form for this purpose. If the address submitted would create a new subscriber, form submission will fail and the error URL will be returned.

An example:

```
<INPUT type="hidden" name="ncreate" value="yes" />
```

 Click to copy


Note: The value of this parameter is not used, only its presence or absence matters.

GFH Examples

GFH Example with Required Fields

Here is an example of a form for the mythical organization named Zanshin Outfitters that contains all required fields.

```
<FORM method="POST" id="example_form" action="https://g001.enterprise.ipost.com/forms.cgi">
<INPUT type="hidden" name="subscribe" value="optin" />
<INPUT type="hidden" name="client_token" value="zanshin_outfitters" />
<INPUT type="hidden" name="success_url" value="http://www.zanshinoutfitters.com/success.html" />
<INPUT type="hidden" name="error_url" value="http://www.zanshinoutfitters.com/error.html" />
Tell us a little about yourself...<br />
Email address:&nbsp;
<INPUT type="text" name="email_addr" size="40" maxlength="255" />
<INPUT type="submit" name="submit" value="Outfit Me" />
</FORM>
```

 Click to copy

This very basic HTML form will save the user's email address to your -hosted database. However the user will **NOT** be opted in to receive your mailings, nor will (s)he be added to any marketing segments. To do these things you must use some of the optional parameters.

GFH Example with Optional Fields

Continuing with the simple Zanshin Outfitters example above, here is an example that contains optional fields.

```
<FORM method="POST" id="example_form" action="https://g001.enterprise.ipost.com/forms.cgi">
<INPUT type="hidden" name="client_token" value="zanshin_outfitters" />
<INPUT type="hidden" name="success_url" value="http://www.zanshinoutfitters.com/success.html" />
<INPUT type="hidden" name="error_url" value="http://www.zanshinoutfitters.com/error.html" />
<input type="hidden" name="listName" value="Spring Contest Entries" />
<h1>Enter the Zanshin Outfitters Spring Contest!</h1>
Tell us a little about yourself...<br />
First name:&nbsp;
<INPUT type="text" name="first_name" size="40" maxlength="255" /><br />
Last name:&nbsp;
```

```

<INPUT type="text" name="last_name" size="40" maxlength="255" /><br />
Zip Code:&nbsp;
<INPUT type="text" name="zipcode" size="10" maxlength="10" /><br />
Email address:&nbsp;
<INPUT type="text" name="email_addr" size="40" maxlength="255" /><br />

Would you like to receive emails from Zanshin Outfitters?<br />
<INPUT type="radio" name="subscribe" value="optin">&nbsp;Yes, please send me
information!<br />
<INPUT type="radio" name="subscribe" value="optout">&nbsp;No, thanks.<br />
<INPUT type="radio" name="subscribe" value="change">&nbsp;I already receive emails from
Zanshin Outfitters but I need to change my address to:<br />
<INPUT type="text" name="new_addr" size="40" maxlength="255" />

Which email format do you prefer?<br />
<INPUT type="radio" name="email_fmt" value="html">&nbsp;HTML<br />
<INPUT type="radio" name="email_fmt" value="text">&nbsp;Text-only<br />

Which topics interest you most?<br />
<INPUT type="checkbox" name="topic_1" value="1" />&nbsp;Camping/Hiking<br />
<INPUT type="checkbox" name="topic_2" value="1" />&nbsp;Biking<br />
<INPUT type="checkbox" name="topic_3" value="1" />&nbsp;Kayaking<br />
<INPUT type="checkbox" name="topic_4" value="1" />&nbsp;Skiing/Snowboarding<br />
<INPUT type="checkbox" name="topic_5" value="1" />&nbsp;Parkour<br />
<INPUT type="checkbox" name="topic_6" value="1" />&nbsp;Sales and Special Promotions<br
/>

<INPUT type="submit" name="submit" value="Outfit Me" />
</FORM>

```

 Click to copy

Encrypted Contact Identity

iPost uses a special type of encrypted token to keep track of the identity of each of your contacts when they interact with your mailings. As a convenient shorthand, this encrypted token is referred to as a **blob**, because it encodes several pieces of information lumped together.


In some cases you may wish to have your form make use of this encoded blob to help manage the data recorded in your database. A specific example is hosting your own subscription form, in which case you may desire to keep track of the relationship between a click in your message and a change in subscription status. There are two steps necessary to accomplish this:

1. In your mailing document, instruct the system to create a suitable blob and provide it to your web server via the URL that references your form. This is done with a {link ...}

expression and the \blob_param attribute: {\link \blob_param=ipost_id
http://www.thisisanexample.com/change}

2. Code the HTML for your form to capture that blob and add it to the action= attribute of your <FORM> tag. This requires either JavaScript, or a server-side dynamic page creation method such as Active Server Pages (ASP). In the example above, when a recipient clicks on the link in the email message the blob is passed to the web server in the ipost_id=... query parameter. The server or JavaScript must find the value of ipost_id and append it to the form action as indicated here:

```
<FORM method="POST" id="example_form" action="https://g001.enterprise.ipost.com/forms.  
cgi/value of ipost_id here">  
Et cetera  
<INPUT type="submit" name="submit" value="Change Your Preference" />  
</FORM>
```

 Click to copy

All of the other form inputs discussed may be passed as usual, but in this case the email_addr input may be omitted, and instead the address is determined by decrypting the blob. *This is the only situation in which email_addr may be omitted.*

Error Conditions

There are several conditions which would result in an error upon submission to the GFH. The GFH will ignore any inputs whose names it does not recognize.

Error Condition	Definition
No client_token field/value	The form does not include an input named client_token or the form was submitted with no value for that input; or, the value does not identify a valid client record; or, the form action includes a blob and the value of client_token does not properly decrypt that blob.
No email_addr field /value	The form action does not contain a blob (see above), and the form does not include an input named email_addr or the form was submitted with no value for that input.
No new_addr field/value (if 'change')	The form was submitted with the subscribe set to a value of change but it either did not include an input named new_addr or that input was submitted without a value.
Invalid email_addr/new_addr value	The address submitted as a value for either email_addr or new_addr was an invalid email address.
Invalid topic_# field/value	The name of a topic input does not match a valid topic

Error Condition	Definition
	ID or the value submitted for that input is invalid.
No success_url	The form does not include an input named success_url.
No error_url	The form does not include an input named error_url.
Unable to Save to the Database	here is a problem saving the data to your hosted database. Please contact your organization's Customer Success Manager.
Feature Not Enabled	The feature has not been enabled on the Database under Settings.

Displaying the Errors

If the webpage defined in the *error_url* includes a DIV with an id of *ipost_error_detail*, the Generic Form Handler will rewrite the contents of that DIV to add an unordered list (UL) of errors which it encountered during processing.

This DIV may be styled or hidden (e.g., in an HTML comment) as you see fit and its inclusion on the error_url webpage is entirely optional.

Retrieving Data

The GFH has a special *fetch* mode which can be used by automated systems to retrieve data for a single customer from your database. This mode returns data in the web standard JSON format. Because of this it is **highly** recommended that this mode be used only by those users with web programming experience. The remainder of this document assumes the reader has that experience.



Fetch mode is currently only available for Lists.

The URL for the fetch mode of the Generic Form Handler is identical to that used for save mode:

<https://g001.enterprise.ipost.com/forms.cgi>

For security reasons, we recommend that you use the POST method for fetching data. The fetch mode of the GFH accepts three parameters, two of which are required; the third is strongly recommended:

Parameter	Definition	Required
fetch	Signifies that the GFH is being used in fetch mode. The value of this parameter is a single email address.	Yes
client_token	The client_token is a unique abbreviation used to represent your client in the system. It is chosen by staff when your client account is set up. The Client Token is found in Client Settings .	Yes
gfh_key	<p>The GFH Key is an encoded string generated when you enable the GFH in the Database Settings. It should be treated as you would treat any confidential password. For example, avoid exposing your GFH Key in HTML sources. You may choose to allow data fetch without a valid GFH Key, but in that case your subscriber data may be fetched by anyone who can guess your client_token and knows the email address of the subscriber.</p> <p>To change your GFH Key or disable key enforcement, disable and then re-enable the Generic Form Handler.</p>	No
success_url	This is the URL for the webpage to which the GFH will redirect the user following a successful submission of the form.	Yes
error_url	This is the URL for the webpage to which the Generic Form Handler will send the user if it encounters any errors while processing the data on the form. The error conditions and how to display the error messages on this webpage are discussed in detail below.	Yes

With very few exceptions (listed in the error conditions below), any other parameters passed to the GFH in fetch mode will be ignored. An example of a correctly formatted URL for a Generic Form Handler fetch request:


```
https://g001.enterprise.ipost.com/forms.cgi?fetch=homer_simpson@ipost.com&
client_token=zanshin_outfitters&gfh_key=9cfbc52d1a762e21562784172e47cec7
```

 Click to copy

That request will return a JSON string something like this (the actual fields returned will depend upon your organization's setup):

```
{ "email_address": "homer_simpson@ipost.
com", "topics": [ { "topic_description": "Catalog", "topic_selected": 1, "topic_id": "1" }, { "topic
_description": "Sales", "topic_selected": 1, "topic_id": "2" }, { "topic_description": "Frequent
Buyer
Progam", "topic_selected": 1, "topic_id": "3" } ], "fields": { "source": "signup_form", "language":
"english", "phone": null, "last_name": "Simpson", "city": "Springfield", "ipost_opt_in_date": "2
003-08-31", "province": null, "ipost_opt_in_detail": "ipost-hosted signup
```

```
form", "postalcode": null, "first_name": "Homer"}, "email_format": null, "opted_out": ""}
```

 Click to copy

This JSON string can then be parsed using any of the standard JSON tools.

Should an error occur in processing, it is returned as an HTTP 404 Status with the specific error in plain text. Errors occur in processing under the following conditions:

- The *fetch* parameter occurs more than once in a single request.
- The *client_token* parameter does not define a valid client record.
- You have chosen to enforce GFH Key checking and the *gfh_key* parameter does not match the key for your client record.
- The email address declared as the value of the *fetch* parameter is not a customer of the specified client or is not a valid email address.
- The parameters *submit* or *btnSubmit* are passed to the Generic Form Handler in fetch mode.
- The Generic Form Handler feature is not enabled for the specified client.