

# Integrating Instamojo

Monday 6<sup>th</sup> April, 2015  
Version 1.2

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Basic features</b>	<b>3</b>
<b>3</b>	<b>Advanced features</b>	<b>6</b>
<b>4</b>	<b>Before a payment</b>	<b>7</b>
4.1	Pre-filling the payment form . . . . .	7
4.2	Passing the amount at the time of the payment . . . . .	7
4.3	Making fields read-only . . . . .	8
4.4	Making fields tamper-proof . . . . .	8
4.5	Passing custom data . . . . .	10
4.6	Hiding custom data . . . . .	11
4.7	Specifying the layout . . . . .	11
4.8	Error codes . . . . .	12
<b>5</b>	<b>After a payment</b>	<b>13</b>
5.1	Redirect . . . . .	13
5.2	Payment API . . . . .	13
5.3	Webhook notification . . . . .	13
5.3.1	How is the webhook different from the redirect? . . . .	14

5.3.2	What comes first, the redirect or the webhook? . . . .	15
5.3.3	How do I test whether webhook notifications are working? . . . . .	15
5.3.4	I'm not getting any webhook requests on my server . .	15
5.3.5	How do I verify that Instamojo is sending me the webhook request . . . . .	16
5.3.6	Should I use the webhook or the API? . . . . .	16
5.3.7	Can you provide a way to simulate a webhook request?	17
<b>6</b>	<b>Instamojo APIs</b>	<b>18</b>
6.1	Payment API . . . . .	18
6.2	Link API . . . . .	18
<b>7</b>	<b>Tools and libraries</b>	<b>19</b>
<b>8</b>	<b>Change Log</b>	<b>20</b>

# 1 Introduction

This document guides you through the process of creating a tighter integration between your web-application and Instamojo.

Should you have any questions about what the most appropriate workflow is for you or for your customers, please feel free to call us at [+91-8880-110-011](tel:+91-8880-110-011) or email us at [support@instamojo.com](mailto:support@instamojo.com).

## 2 Basic features

There are several ways for you and your customers to interact with Instamojo. The simplest way is for you to create and share an Instamojo link with your customers.

Instamojo links support a host of very powerful features. Please review and take advantage of these features before considering our more advanced features, described in the next section.

**Smart descriptions** The description section of your payment link supports a subset<sup>1</sup> of Markdown<sup>2</sup>, that allows you to add beautiful formatting to the text you add. Also, if you add a link to media content from selected popular rich media sites<sup>3</sup> on its own line, the link is automatically converted into an embed. No need to fumble around and search for embed code.

Example: <https://www.instamojo.com/instamojo/integration-documentation/>

**Custom fields** These allow you to specify extra fields of data that you would like your user to enter during the checkout process. The default fields available in a payment link are the name, the email address and the phone number of the buyer. You may request for more information, such as the company name, age, etc. according to your needs.

Example: <https://www.instamojo.com/demo/demo-custom-fields/>

---

<sup>1</sup>Not all features of Markdown are supported. In particular, any HTML or Javascript added will be removed

<sup>2</sup>Syntax: <http://daringfireball.net/projects/markdown/syntax>

<sup>3</sup>This includes services like Youtube, Flickr, Vimeo, Slideshare, Scribd, Instagram, SoundCloud, Twitter and many more

**Discount codes** These allow you to provide some of your users with discount codes. You have the option to render a discount code box on your payment link where users can enter their code manually (Eg: <https://www.instamojo.com/demo/demo-payment-features/>) or, you can share a link with a discount directly with your users (Eg: <https://www.instamojo.com/demo/demo-payment-features/?discount=off15>).

**Variants** If you have multiple variants/tiers/options/colours of the same product or service that you're selling, you don't need to create individual links for each of them. You can add these variants to the same Instamojo link. You can specify limits to the quantity of each variant being sold, if you choose.

Example: <https://www.instamojo.com/demo/demo-payment-features/>

**Custom notes** You can add a custom note to a payment link. This note is only shown to a user at the end of a successful purchase. This is displayed in the post-purchase success page and in the email that is sent to the buyer after a purchase. You can include a thank-you message, instructions to get started, any keys or passwords required, etc.

Example: <https://www.instamojo.com/instamojo/integration-documentation/>

**Quantity limits** You can restrict the number of items that can be purchased for a given link.

**Socialpay** If you'd like to give something away for free, you can accept social payments in the form of a Tweet or Facebook post.

Example: <https://www.instamojo.com/instamojo/integration-documentation/>

**Digital downloads** If you'd like to sell your music, e-books, or other digital content, you can take advantage of our *Digital Goods* links. You can upload the file to our servers and we take care of hosting, bandwidth, user downloads, future retrievals, etc.

Example: <https://www.instamojo.com/instamojo/integration-documentation/>

**Affiliate system** We have an affiliate system in place where you can invite your affiliates to help you increase your sales. We'll handle their payouts and KYC procedures so you don't have to.

We offer all of the above features and many more that are being added every day. If you have any questions about whether we can support your

use cases or workflow, or if you're having difficulties finding any of the above features in your account, please do not hesitate to reach out to the Instamojo team.

### 3 Advanced features

If the features in the previous section do not meet your needs, or if you need a tighter integration between your website and Instamojo, we provide several integration options:

**Before a payment** Before a customer reaches an Instamojo page, you can specify the behaviour of the Instamojo page through URL query-string parameters.

- *Pre-fill* any or all of the fields in the payment form.
- Make any of the fields *read-only* and *tamper-proof*.
- If it isn't a fixed price payment, pass the *amount* at the time of the payment.
- Send us any *custom data* (such as an order ID or a username) that you'd like us to *send back* after the payment.
- Specify the *layout* of the Instamojo page.

**After a payment** After a payment successfully ends, there are several ways in which you can choose to be notified:

- We can *redirect* the customer to a URL that you specify.
- You can query our *API* to get details about a specific transaction (read below).
- We can send you a *webhook notification* to your server.

**Instamojo APIs** We have the following APIs that permit programmatic access:

- *Payment API*: Get details about a specific payment or all your payments programmatically.
- *Link API*: View, create or edit links in your account programmatically.

## 4 Before a payment

There are several ways in which you can customize a payment link using URL query-string parameters. Please do not forget to URL encode any parameters you send.

### 4.1 Pre-filling the payment form

Every Instamojo link contains a payment form that will contain, at the very least, fields to ask for the buyer's *name*, *email address* and *phone number*. If you have these details before they arrive on this page, you can send the values across so that they don't have to fill it up again.

Field	Key	Character limit
Name	<code>data_name</code>	20
Email address	<code>data_email</code>	75
Phone number	<code>data_phone</code>	20

Example:

[https://www.instamojo.com/demo/demo-offer/?data\\_name=Aditya+Sengupta&data\\_email=aditya@instamojo.com&data\\_phone=9821485060](https://www.instamojo.com/demo/demo-offer/?data_name=Aditya+Sengupta&data_email=aditya@instamojo.com&data_phone=9821485060)

### 4.2 Passing the amount at the time of the payment

By default, links on Instamojo have a fixed price. If you would like to pass the amount at the time the transaction is taking place, you'll need to first edit the link and make the following changes:

1. Ensure the price is greater than 0 (say Rs. 10).
2. Enable the “*pay what you want*” option.

Once you have done this, you can pass the price using the following URL query parameter:

Field	Key	Character limit
Amount	<code>data_amount</code>	Any number up to 2 decimal places

Example:

[https://www.instamojo.com/demo/demo-offer/?data\\_amount=123.45](https://www.instamojo.com/demo/demo-offer/?data_amount=123.45)

## 4.3 Making fields read-only

**Note:** This section only describes how to specify read-only fields. Fields specified as read-only will have the HTML `readonly` attribute added to them. A savvy user can still edit the URL or the HTML and change the values being sent. Please read section 4.4 to find out how to make a read-only field tamper proof.

You can make any of the payment form fields read-only by passing the name of that field as a value to the key `data_readonly`. If there are multiple fields you would like to make read-only, simply repeat this for every field (in the same way you would use an HTML select-multiple field<sup>4</sup>).

Example:

```
https://www.instamojo.com/demo/demo-offer/?data_name=Aditya+Sengupta&data_email=aditya@instamojo.com&data_readonly=data_name&data_readonly=data_email
```

## 4.4 Making fields tamper-proof

To ensure that your read-only fields cannot be tampered with, you can sign your links using the procedure below.

**Note 1:** To do this, you'll need the **salt** for your Instamojo account. You can get this by logging into your Instamojo account and visiting the [Developers page](#).

**Note 2:** You will need to contact the Instamojo team to enable permanent tamper-proofing for your link. If you do not, anyone can remove the URL query-string parameters and be able to make a payment with a value that is less than what you expect. Once permanent tamper-proofing is enabled, we will refuse to accept payments on links that are not signed.

For the purpose of this example, we assume you're trying to make the following link tamper-proof:

```
https://www.instamojo.com/demo/demo-offer/?data_name=Aditya+Sengupta&data_email=aditya@instamojo.com&data_phone=9821485060&data_amount=123.45&data_readonly=data_name&data_readonly=data_email&data_readonly=data_phone&data_readonly=data_amount
```

---

<sup>4</sup>HTML select-multiple field: [http://www.w3schools.com/tags/att\\_select\\_multiple.asp](http://www.w3schools.com/tags/att_select_multiple.asp)



1. Arrange the **read-only** fields in the alphabetical order of their keys. If you have any keys with upper-case letters, convert them to lower-case letters first.

In this example, you would get the following order:

- (a) `data_amount`
- (b) `data_email`
- (c) `data_name`
- (d) `data_phone`

2. Using the order above, replace the keys by their respective values.

In this example, you would get the values below in the following order:

- (a) `123.45`
- (b) `aditya@instamojo.com`
- (c) `Aditya Sengupta`
- (d) `9821485060`

3. Concatenate the above values into a single string, with each value separated by a pipe character, i.e, the `|` character.

Using the above example, you get the following string:

`123.45|aditya@instamojo.com|Aditya Sengupta|9821485060`

4. Use the above string as the message for the HMAC-SHA1 algorithm<sup>5</sup> and the salt for your Instamojo account as the salt for the algorithm. The output of this will be the signature we need.

For example, if your salt is “abcde”, the signature you would generate using the string from the previous step as the message is:

`6f905be9811990707f9d833da8e93bfebb23abbc`

Once you have the signature using the above procedure, you add it as the value of the `data_sign` key in the URL. The URL would then be:

---

<sup>5</sup>Hash-based Message Authentication Code: [http://en.wikipedia.org/wiki/Hash-based\\_message\\_authentication\\_code](http://en.wikipedia.org/wiki/Hash-based_message_authentication_code)

[https://www.instamojo.com/demo/demo-offer/?data\\_readonly=data\\_name&data\\_readonly=data\\_email&data\\_readonly=data\\_phone&data\\_readonly=data\\_amount&data\\_sign=6f905be9811990707f9d833da8e93bfebb23abbc&data\\_email=aditya@instamojo.com&data\\_amount=123.45&data\\_name=Aditya+Sengupta&data\\_phone=9821485060](https://www.instamojo.com/demo/demo-offer/?data_readonly=data_name&data_readonly=data_email&data_readonly=data_phone&data_readonly=data_amount&data_sign=6f905be9811990707f9d833da8e93bfebb23abbc&data_email=aditya@instamojo.com&data_amount=123.45&data_name=Aditya+Sengupta&data_phone=9821485060)

Note that the above URL will not actually work since the salt for the demo account is not actually “abcde”.

Don’t forget to URL encode the query-string parameters!

## 4.5 Passing custom data

You may want to pass custom data to Instamojo at the beginning of a payment that you would like to retrieve at the end of a payment, such as an order ID, or a username of someone on your application.

1. Open your Instamojo dashboard and search for the link you would like to add custom data to. (Don’t click the link).
2. Below the link, you’ll find the option to add and edit “*Custom Fields*”. Click on this.
3. Add the appropriate fields. You can add as many as you like.
4. For each field that you would like to pass the data for in the URL, you’ll need the identifier of that field. You can find the identifier by using the mouse to hover over the field name you’ve just created and the identifier will appear. It will look like `Field_<number>`, where `<number>` will be a number. Note down the identifiers for each custom field you define.
5. The data key for the query-string parameter will be generated by prefacing the identifier with `data_`. For example, if the identifier is `Field_12345`, the data key would be `data_Field_12345`. Note the placement of the underscores and the case sensitivity.
6. You can add your custom data using the data keys above. Here is an example:

[https://www.instamojo.com/demo/demo-custom-fields/?data\\_Field\\_70240=27&data\\_Field\\_70241=Instamojo&intent=buy](https://www.instamojo.com/demo/demo-custom-fields/?data_Field_70240=27&data_Field_70241=Instamojo&intent=buy)

Note that custom data fields can be made read-only and signed using the same procedure described in section 4.4.

## 4.6 Hiding custom data

You may want to hide the custom data (such as order IDs or internal usernames) that is displayed in the payment form, especially if it is not of immediate obvious use to the buyer. To do this:

1. Add the key `data_hidden` as a query-string parameter to the URL.
2. Use the data key of the custom field as the value of the `data_hidden` parameter.
3. Repeat for all the custom fields you want to hide.

Example: [https://www.instamojo.com/demo/demo-custom-fields/?data\\_Field\\_70240=27&data\\_Field\\_70241=Instamojo&intent=buy&data\\_hidden=data\\_Field\\_70240&data\\_hidden=data\\_Field\\_70241](https://www.instamojo.com/demo/demo-custom-fields/?data_Field_70240=27&data_Field_70241=Instamojo&intent=buy&data_hidden=data_Field_70240&data_hidden=data_Field_70241)

## 4.7 Specifying the layout

There are four layouts that are provided by Instamojo.

**Default layout** This is the default layout you see when you open an Instamojo link.

**Payment button** You can make the Instamojo payment link appear as part of your web-application by using our payment button. You can customize your payment button and generate the embed code by searching for your Instamojo link in your dashboard and clicking on the “*Payment button*” option below your link.

The “*Redirect*” style will simply open the Instamojo link in a new tab and allow the payment to take place there.

The “*Remote checkout*” style will open a modal overlay on your web-application and allow the payment process to seamlessly integrate with your site.

**Form open** If you’re pre-filling the payment form, you may want to keep the payment form open by default, so the buyer is saved an extra click of having to open the form. You can do this by adding the query-string parameter, `intent=buy` to your Instamojo link.

Example: <https://www.instamojo.com/demo/demo-offer/?intent=buy>

**Minimal link** If you want to embed the Instamojo link in an iframe, you may want a minimal version of the Instamojo page to appear. To do this, simply add the query-string parameter, **embed=form** to your Instamojo link and you'll get a skinned version of the Instamojo link that consists only of the payment forms.

Example: <https://www.instamojo.com/demo/demo-offer/?embed=form>

## 4.8 Error codes

If you've made an error with one of the query parameters, you may see one or more of the following error codes:

Error code	Error description
QF1	Name is not valid
QF2	Email is not valid
QF3	Phone is not valid
QF4	Amount is not valid
QF5	Readonly fields are not valid
QF6	Hidden fields are not valid
QF7	Signature is not valid
QF10	Custom field is not valid
QF11	Signature not specified for link that requires signature
QF12	Readonly fields not specified for link that requires signature
QF13	Data to sign invalid for link that requires signature
QF14	Data to sign not specified for link that requires signature
QF15	Signature not valid

## 5 After a payment

There are several ways in which you can process a successful transaction.

### 5.1 Redirect

At the end of each transaction on a given link, the user can be redirected to a link that you specify. You can specify this link in the “*Advanced settings*” section while creating or editing a link. While we redirect the user, we also add the payment ID as a query parameter (using the key `payment_id`) to your redirect link.

You can use the payment ID to query our Payment API (described in section 6.1 on page 18) to verify the status of the payment and retrieve other details, such as the amount, the custom data, the buyer’s details, etc. More on this in the next section.

### 5.2 Payment API

The payment API allows you to do the following:

1. Retrieve a list of payments associated with your Instamojo account.
2. Given a payment ID, retrieve the data associated with a specific payment.

The details of this are provided in section 6.1 on page 18.

### 5.3 Webhook notification

At the end of a successful payment, we can initiate a server to server POST request that sends you all the data associated with that payment. You can specify the URL to which we send the webhook request in the “*Advanced settings*” section while creating or editing a link. We send the following data with each webhook request:

The following parameters are always sent:

Variable	Key
Payment ID	payment_id
Status	status
Slug of the link	offer_slug
Title of the link	offer_title
Buyer's email address	buyer
Buyer's name	buyer_name
Buyer's phone number	buyer_phone
Currency	currency
Quantity of items purchased	quantity
Price of each item	unit_price
Total amount paid	amount
Fees charged by Instamojo	fees
Message authentication code	mac

The following parameters are only sent when applicable:

Variable	Key
Shipping address specified	shipping_address
Shipping city	shipping_city
Shipping state	shipping_state
Shipping postal code	shipping_zip
Shipping country	shipping_country
Discount code used	discount_code
Amount discounted	discount_amount_off
Variant options specified	variants
Custom fields (JSON blob)	custom_fields

### 5.3.1 How is the webhook different from the redirect?

A webhook notification is an HTTP *POST* request that is made from our server. A redirect is an HTTP *GET* request that is made from the buyer's web browser.

In a webhook notification, we send along all the data of the payment. In a redirect, we only send the payment ID of the transaction as a URL query-string parameter. The redirect is typically used in conjunction with the API (by making use of the payment ID as the key to poll the payment API with) while the webhook is often used independently.

### 5.3.2 What comes first, the redirect or the webhook?

The webhook request is asynchronous with the redirect. The redirect takes place on the user's browser while the webhook request is initiated from our server. Therefore, there is no guarantee that you will receive one before the other.

### 5.3.3 How do I test whether webhook notifications are working?

To check whether webhook requests are being sent from Instamojo's servers or to see what data is being sent through the webhook request, you can use [requestb.in](https://requestb.in). Simply create a new Requestb.in URL and paste the URL in the webhook URL field of your Instamojo link.

To view the data sent to the webhook URL, open the Requestb.in URL in your browser after adding `?inspect` to the URL. For instance, if the URL you enter in the webhook field is `http://requestb.in/example`, you'll need to open `http://requestb.in/example?inspect` in your browser to view requests sent to this URL. Of course, to send data to this URL, simply make a transaction on the Instamojo link that you've set up with this Requestb.in URL.

Remember that webhooks work with free links too, so you can create a link priced at Rs. 0 for testing purposes and it'll still work.

### 5.3.4 I'm not getting any webhook requests on my server

There are several reasons for which you may not be receiving any webhook requests.

- Webhooks requests are made through a deferred task queue, so they may not fire immediately as a payment happens. There may be a delay of a few seconds if our servers are under heavy task loads. You can read [5.3.3](#) to verify whether the webhook requests are being made from Instamojo's servers.
- Certain web servers block out requests made using non-browser query strings. You'll need to whitelist the user agent that Instamojo uses to send webhook requests if this is the case.
- Our webhook requests currently time out after 5 seconds, if no response is received from your server.

- You may not have configured your webhook receiver correctly. Read [5.3.7](#) to see how to simulate a webhook request.
- Our webhook request need a publicly accessible URL to work. They won't work with localhost URLs and local IP address ranges.

### 5.3.5 How do I verify that Instamojo is sending me the webhook request

Every webhook request includes a field named `mac`, which is the message authentication code. This code is generated using the content of the message and a secret salt that is accessible only to you and to Instamojo. You should check every webhook request against the message authentication code included and verify that it is the same as the code that you are able to generate using the content of the message and your salt<sup>6</sup>.

The procedure for calculating the webhook MAC is similar to the procedure described in section 4.4. Simply use all the fields of the message except the `mac` field itself for generating the message authentication code and verify it against the one you've received<sup>7</sup>.

### 5.3.6 Should I use the webhook or the API?

To decide whether you should use the webhook or the API, you'll need to keep in mind that webhooks are asynchronous and deferred.

- If you need the transaction details immediately after redirecting the buyer to your site, then you'll need to use the API.
- If you need to show a custom page to the buyer after the payment that depends on the transaction details, then you'll need to use the API.
- If you only need to update your database or send an email to the buyer after a transaction, then you can use the webhook.

---

<sup>6</sup>You'll find your salt here once you log into your Instamojo account:  
<https://www.instamojo.com/developers/>

<sup>7</sup>The procedure is described in detail here:  
<http://support.instamojo.com/support/solutions/articles/139575-what-is-the-message-authentication-code-in>



### 5.3.7 Can you provide a way to simulate a webhook request?

In order to simulate a webhook request by yourself for testing or debugging purposes, you can issue the following curl request:

```
curl --data "param1=val1&param2=val2" http://your-webhook-url.com/endpoint/
```

## 6 Instamojo APIs

Instamojo primarily provides the two following REST APIs- the Payment API and the Link API. To use either of these APIs, you'll need a private API key and auth token. You can view these credentials by logging into your Instamojo account and then visiting <https://www.instamojo.com/developers/>.

Documentation for the Instamojo APIs is provided at <https://www.instamojo.com/developers/rest/>. A summary is provided below.

### 6.1 Payment API

This API can be used to retrieve all the details of a specific payment or a list of payments made to your Instamojo account. This is typically used at the end of a payment to verify the status of a payment. You can retrieve a payment ID from a redirected URL as described in section 5.1 on page 13.

**Note:** You must verify the status of every payment using the API or by logging into your Instamojo account if you're using a redirect based workflow to check whether a payment is successful or not.

### 6.2 Link API

This API can be used to create and edit links in your Instamojo account. This is typically used when you need to upload a large number of products to your Instamojo account.

If you have any questions on which API you should use, please don't hesitate to reach out to the Instamojo team.

## 7 Tools and libraries

**Python API wrapper** This is an implementation of the Instamojo REST API in Python.

<https://github.com/Instamojo/instamojo-py>

**PHP API wrapper** This is an implementation of the Instamojo REST API in PHP.

<https://github.com/Instamojo/instamojo-php>

**Prestashop plugin** If you use Prestashop for your shopping cart needs, then this plugin allows you to set up Instamojo for checkout.

<https://github.com/Instamojo/instamojo-prestashop>

**Wordpress plugin** This is a sample Wordpress plugin and can be used to showcase your Instamojo links on Wordpress.

<https://github.com/Instamojo/instamojo-wordpress-plugin>

**IM Tools** This repository contains a few useful Instamojo tools- such as a sample Webhook receiver and a command line tool to generate pre-filled and signed links on Instamojo. To use the link tool, simply download the Python file to your computer, open a command line shell and execute the following command for a tutorial on how to use the tool:

```
python link.py -h
```

<https://github.com/Instamojo/instamojo-wordpress-plugin>

## 8 Change Log

### **Version 1.1: November 11, 2014**

The first version of this document was released.

### **Version 1.2: April 6, 2014**

1. Updated the support phone number.
2. Fixed an incorrect URL in an example.
3. Added a cautionary note for aforementioned example.
4. Added this change log and document version number.