



PAY FOR FUN

GO API (GATEWAY) – VERSION 1.1.0

(download the latest version: <https://api.p4f.com/developer/1.0/getgoapidocument/>)

TABLE OF CONTENTS

Contents

Considerations _____	1
Introduction _____	2
API Pay4Fun GO _____	3
Planned Maintenance _____	4
API Endpoints _____	5
Web requests quota control per IP _____	6
Supported Currencies _____	7
Obtaining Merchant's Account Credentials _____	8
Understanding Merchant's Account Credentials _____	9
Whitelisting Pay4Fun's Servers IPs _____	10
Pay In _____	11
Pay in Flow Overview _____	12
Pay in Flow Explained _____	13
Pay In Request _____	14
Pay In Response _____	20
Pay In Checklist _____	22
Pay In Confirmation _____	23
Pay In Confirmation Checklist _____	27
Pay In Query Status _____	28
Pay In Query Status Checklist _____	33
Payment Methods _____	34
Appendix 1 – Generating the Hash or Sign _____	36
Appendix 2 – Error Messages _____	38
Appendix 3 – Merchant Labels _____	40
Appendix 4 – Security issues using iframes _____	41

TABLE OF CONTENTS

Appendix 5 – Pay4Fun Logo	42
---------------------------	----

Considerations

WARNING

The information within this document is subject to change without notice.

The software described in this document is provided under a license agreement and may be used or copied only in accordance with this agreement.

No part of this manual may be reproduced or transferred in any form or by any means without the express written consent of Pay4Fun. All other names, trademarks, and registered trademarks are the property of their respective owners.

P4F makes no warranty, either express or implied, with respect to this product, its merchantability or fitness for a particular purpose, other than as expressly provided in the license agreement of this product. For further information, please contact us.

Introduction

PAY4FUN GO

Welcome to Pay4Fun, we allow consumers and merchants to make transactions quickly, conveniently and safely.

This API document is for Pay4Fun GO, a gateway system for direct deposit into the merchant website.

Make sure you also integrate Pay4Fun (e-wallet) for more payment options to your customer. Please refer to the Pay4Fun (e-wallet) API document on <https://api.p4f.com/developer/1.0/GetGoApiDocument/>

API Pay4Fun GO

Pay4Fun's API works with a **redirect integration model**: you will redirect customers from your website to the Pay4Fun Authentication page where they type their Pay4Fun's credentials to complete the payment. Pay4Fun is responsible for:

- Validating the data entered by the customer;
- Ensuring that the data is held and disposed securely;

Important

Due security matters, **loading any Pay4Fun URL inside an iframe is not allowed!** Please check Appendix 4 – Security issues using iframes for more information.

This model allows you to accept payments from Pay4Fun, without adding complexity (such as handling payment data) to your own system. With the redirect model, you can also choose to:

- Specify your website URLs that redirect the customers after they complete the payment (success or fail);
- Dynamically control the minimum or maximum amount of payments that you want to offer;

Planned Maintenance

Occasionally, Pay4Fun will have to cause short periods of downtime in order to perform a planned maintenance; a proactive approach to execute needed actions in order to provide the maximum level of security and performance for our customers.

During such short periods of time, it will not be possible to login at merchant's back-office and all API calls will return the following error:

Response code 503 - Service Unavailable

Response body

```
"Retry-After 2019-04-23 21:16:25 GMT"
```

Response headers

```
"content-type: application/json; charset=utf-8"  
"date: Tue, 23 Apr 2019 21:11:25 GMT"  
"retry-after: 2019-04-23 21:16:25 GMT"
```

Notice that the response header contains the GMT date and time (retry-after) which the planned maintenance is expected to end. You should gracefully handle this error at your side and wait until the maintenance period ends.

Important

No transaction will be processed or even stored at Pay4Fun's side during a planned maintenance.

API Endpoints

The P4F RESTful API, where all responses are in JSON, is available in the P4F Sandbox environment for integration testing purposes. To switch between the Sandbox and the live production system you only need to change the endpoint URI and the credentials.

The main URI **<P4F_API>** is defined below for Sandbox and Production environments. Please, check carefully the method explanation to use the right URI for each request:

Warning - Sandbox URL (Testing):

- API: **<http://apitest.p4f.com>**

Important – Production URL (Live stage)

- API: **<https://api.p4f.com>**

Web requests quota control per IP

Pay4Fun's API has a feature to control quota of web requests per IP; it means that requests originated from the same IP will be blocked if quota limits are reached.

The quota control has limits for different endpoints allowing an IP to make a maximum number of calls in a time interval.

The quota control exists to ensure the service is up and running for all Merchants, avoiding API running out of response capacity due to misuse of features or malicious requests.

The actual quota limits are specified below:

REQUESTS QUOTA CONTROL

API endpoint	HTTP verb	Quota (requests per second)
/1.0/go/process	POST	6
/1.0/go/transaction	POST	3
* (any other endpoints together)	POST	6

These quota limits above are not customizable, they are the same for every Merchant.

Please, note that these limits may change without prior notice.

SUPPORTED CURRENCIES

Supported Currencies

The following currencies are supported by P4F's API:

SUPPORTED CURRENCIES

ISO 4217 code	Name
BRL	Brazilian Real
GBP	British Pound
USD	United States Dollar
EUR	European Euro

Transactions will be processed according the specified currency at the request.

Make sure to send the correct information to the API. Pay4Fun takes no responsibility for incorrect amount or currency parameters sent to the API.

Obtaining Merchant's Account Credentials

After your account has been successfully approved by P4F, you will receive credentials to access the back-office (user name and password) where you will be able to retrieve the account credentials needed to configure the API requests accordingly.

Warning - Sandbox URL (Testing):

- <http://backtest.p4f.com/auth/merchant/login/>

Important – Production URL (Live stage)

- <https://back.p4f.com/auth/merchant/login/>

At the menu **ACCOUNT / API / CREDENTIALS**, subsection **DIRECT API**, you will find:

- Account Credentials:
 - **Merchant ID;**
 - **Merchant Key;**
 - **Merchant Secret;**
- Latest version of API document;
- Integration helper kit with JSON API requests;

Understanding Merchant's Account Credentials

Account credentials are available in your back-office for both staging and production environment. These credentials consist of:

MID (Merchant ID) - This is a unique identifier provided to every merchant by Pay4Fun. MID is part of your account credentials and is different on staging and production environment.

Merchant Key - This is a private key used for secure encryption of every request. This needs to be kept on server side and should not be shared with anyone.

Merchant Secret - This is a secret piece of information used as part of the seed string to generate communication hash. This needs to be kept on server side and should not be shared with anyone.

Important

Merchant secret and **Merchant key** are sensitive data and should be kept secure and safe from non-authorized personal and never should be included as plain text in any communication, electronic or not!

Whitelisting Pay4Fun's Servers IPs

In case you keep an IP whitelist to authorize access to your server, you must add Pay4Funs IPs to it.

Accessing the Merchants back-office, under the menu **ACCOUNT / API / CREDENTIALS**, you can check the IP list to be whitelisted for Pay4Fun's Test and Live stages.

Important

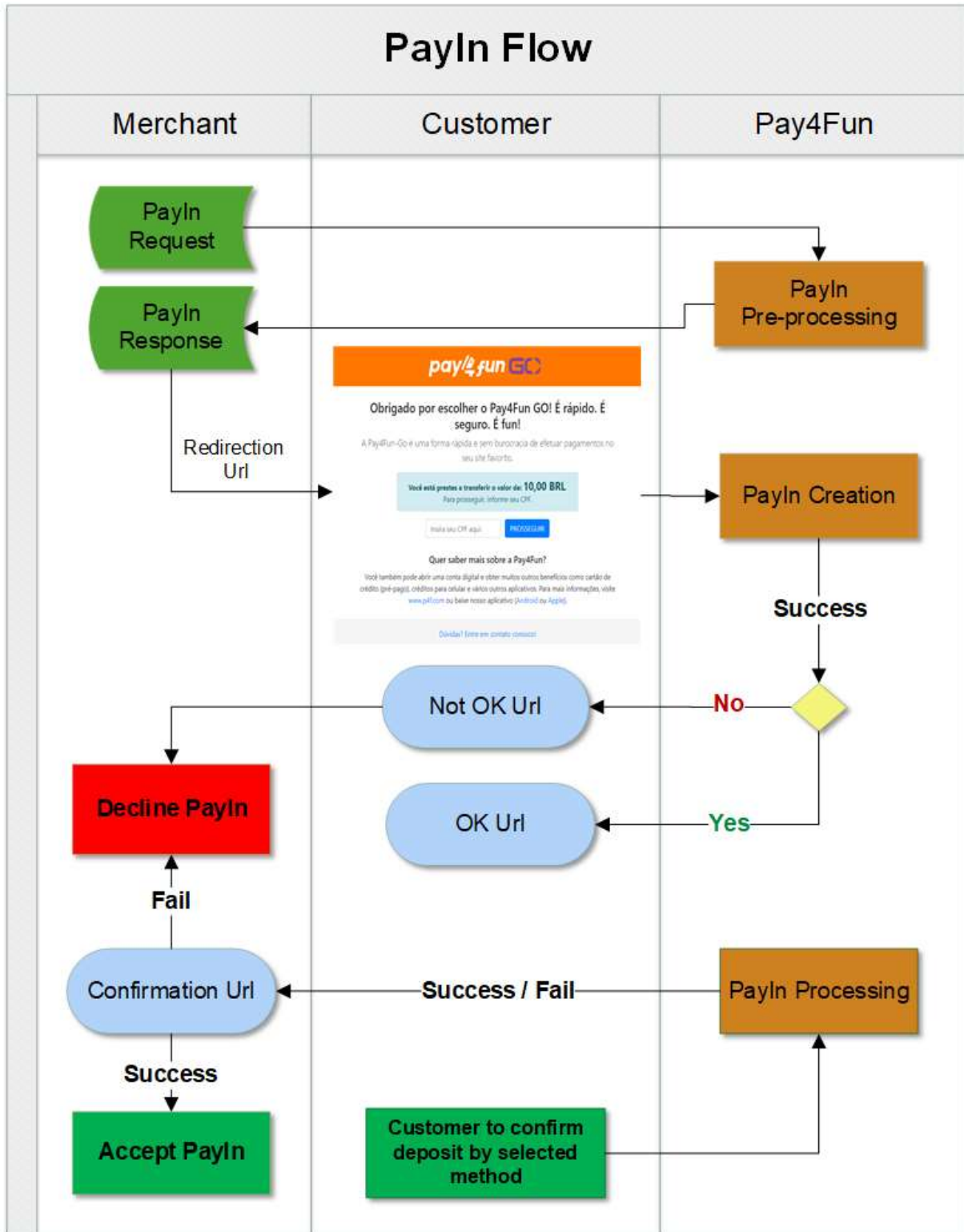
Not adding the IPs to your whitelist will avoid the API's confirmations to reach your servers.

Pay In

A Pay In transaction occurs when a Pay4Fun Customer transfers from his/hers wallet into the Merchant wallet.

Minimum and maximum limits are to be defined by the Merchant but must be between BRL20.00 and BRL30,000.00. Maximum amount varies according to the deposit method selected. All transactions (and maximum limit) are subject to verification by Pay4Fun due to KYC and AML-CFT policies, and might be declined.

Pay in Flow Overview



Pay in Flow Explained

TRANSACTION CREATION

When a transaction request is received at Pa4Fun's server, there are multiple validations carried out like valid source of request, structure of request, uniqueness of Merchant Invoice ID, etc. Once these validations are passed, a transaction is created.

SUCCESSFUL TRANSACTION

Customer fills authentication details to authorize the payment. Once the authorization is successful, money is credited to merchant's account. This transaction is a successful transaction.

FAILED TRANSACTION

If for any reason the transaction processing fails, **the transaction is not created at Pay4Fun's system** and merchant will be notified through the NOT OK UL specified at the API's request.

PENDING TRANSACTION

The transaction is waiting for Customer's payment.

Pay In Request

The purpose of this request is allow your Customers authenticate their Pay4Fun's wallet.

Important

No transaction will be generated due this request!!

You will receive an URL within the response and you should redirect your Customer to this URL where his authentication will take place.

The request header should contain two parameters: **merchantId** and **hash** for security and authorization matters.

Please, be careful to use the correct URL:

Sandbox (Testing):

<P4F_API>: <http://apitest.p4f.com>

Production (Live stage)

<P4F_API>: <https://api.p4f.com>

PAY IN REQUEST

PAY IN API REQUEST:

POST <P4F_API>/1.0/go/process/

Header

```
{
  merchantId: 123456,
  hash: "xxxxxxxxxx"
  Content-Type: "application/json"
}
```

An example of a valid Pay In request is shown below:

```
{
  "amount": 10.00,
  "merchantInvoiceId": "ABC777",
  "language": "en-US",
  "currency": "BRL",
  "okUrl": "https://merchant.com/ok_url",
  "notOkUrl": "https://merchant.com/not_ok_url",
  "confirmationUrl": "https://merchant.com/confirmation_url"
}
```

REQUEST PARAMETERS SPECIFICATION

Parameter	Type	Description	Mandatory
hash	String	Hash to avoid fraud (check Appendix 1 – Generating the Hash section)	Yes
amount	Decimal	Pay in amount (2 digits after the decimal point)	Yes
merchantInvoiceId	String	Merchant's transaction unique identifier (maximum 250 characters)	Yes
language	String	Customer's desired language (pt-BR, en-US or es-ES)	No
currency	String	Pay in currency (ISO 4217 - 3 characters code)	Yes
okUrl	String	Merchant's redirection URL for successful transactions	Yes
notOkUrl	String	Merchant's redirection URL for unsuccessful transactions	Yes
confirmationUrl	String	Merchant's confirmation URL	Yes
labelId	Integer	If a valid label ID is informed, transaction will be tagged. Please check Appendix 3 – Merchant Labels for more details	No
Onboard Data parameters (check * below)			
zipcode	String	Customer's Brazilian postal code	No
email	String	Customer's e-mail	No
birthDate	String	Customer's birthdate (format yyyy-MM-dd, e.g., 2000-03-30)	No

PAY IN REQUEST

fullName	String	Customer's full name	No
Payment Method parameters (check ** below)			
paymentMethod	String	Desired Payment Method Type: PIX , Boleto or BankTransfer	No
paymentMethodId	Integer	Check Payment Methods section for more details	No

* ONBOARD PARAMETERS

To facilitate the user experience, you have the option to inform personal data from your customer.

If you decide to send, we will treat on our end and populate the necessary fields during the process. If you decide not to send, we will request the customer to fill in the fields.

We will treat only the first request of the customer. If you keep sending onwards we will disregard these data, so feel free to send it only once.

Parameters are: postal code, e-mail address, date of birth and full name.

** PAYMENT METHOD PARAMETERS

When offering Pay4Fun GO on your deposit page, you have the option to present it as Pay4Fun GO (one icon) and/or one icon for each different payment method.

So when using the Pay4Fun GO icon you should NOT inform any of the Payment Method parameters in the request. We will show the available payment methods so customer can decide which one to use.

PAY IN REQUEST

When using the icon for each payment method, use ONE of the following parameters in the request: `paymentMethod` or `paymentMethodId`. We will redirect and show the confirmation screen for the selected payment method.

PAY IN HASH SEED STRING

Concatenate the following information, in that order, to obtain the seed string for hash generation for Pay in requests (check Appendix 1 – Generating the Hash section):

- Merchant's ID
- Pay in amount in cents (comma or decimal point **with two decimal places**)
- Merchant's invoice ID
- Merchant's secret

Consider a Pay in request of **10.00 USD** related to the merchant's invoice ID **ABC777**.

Assuming merchant's ID is **34567** and merchant's secret is "**abcdef**", the seed string for hash generation would be (in that order):

1. merchant's ID;
2. transaction's amount with cents;
3. merchant's invoice ID;
4. merchant's secret;

Seed string = **345671000ABC777abcdef**

The Merchant's secret should be kept safe because is the secret factor that prevents others to generate valid hash.

Important

Even in cases of **flat amounts**, for instance, 1000.00, **cents must be included at the seed string**.

The P4F API will generate a hash for each request and compare to the Merchant's hash. Any attempt of forging a request without the knowledge of the Merchant's secret will lead to an invalid hash and the transaction will be declined.

Pay In Response

SUCCESS RESPONSE

In case of successful response you will receive a response containing an URL to redirect your customer to the P4F's authentication page.

After providing the authentication credentials - a valid CPF (Brazilian personal ID number) – the Pay in processing will start.

If the authentication went well, you should receive a **successful response** as below:

```
{
  "code": 201,
  "message": "success message",
  "url": "<P4F_API>/1.0/go/auth/<KEY>"
}
```

You must redirect your customer to the URL where the Pay in processing will occur, and the transaction performed.

RESPONSE PARAMETERS SPECIFICATION

Parameter	Type	Description
code	Integer	HTTP response code
message	String	Response message
url	String	URL to redirect Customer to be authenticated by P4F and confirm the transaction

UNSUCCESSFUL RESPONSE

An example of a **fail response**:

```
{  
  "code": 400,  
  "message": "error message"  
}
```

If any detail prevents transaction being completed, API will redirect your customer to your Not Ok URL.

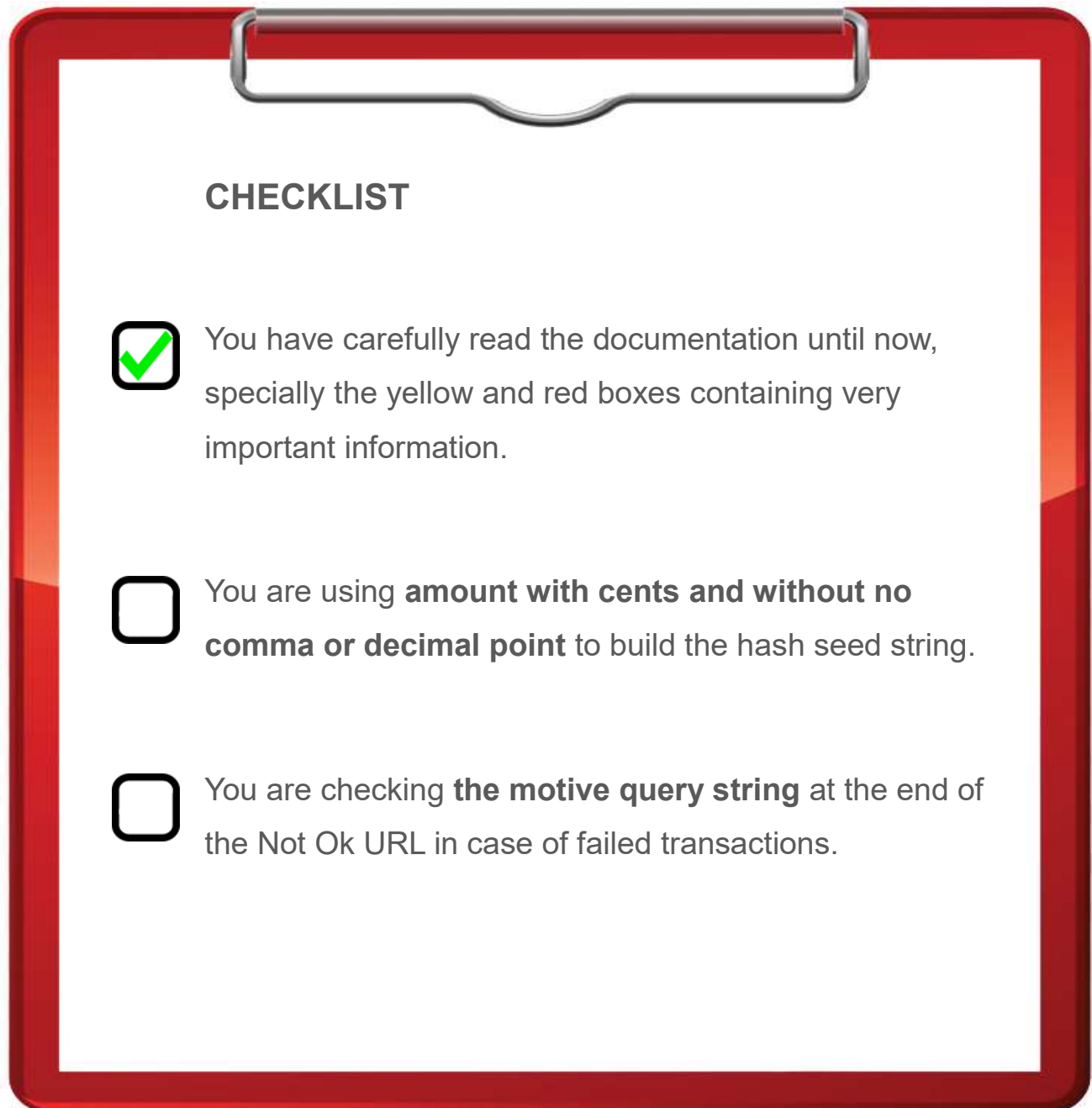
Important

Please, notice that the **motive of the failure will always be added to the end of your Not Ok URL as query string** as shown below:

`http://yourNotOkUrl?motive=invalid_request`

Pay In Checklist

Before going any further and definitely before the Go Live, we suggest you go through this highly recommend checklist to confirm the integration is ready to go and avoid problems in the future:



CHECKLIST

- You have carefully read the documentation until now, specially the yellow and red boxes containing very important information.
- You are using **amount with cents and without no comma or decimal point** to build the hash seed string.
- You are checking **the motive query string** at the end of the Not Ok URL in case of failed transactions.

Pay In Confirmation

Important:

You must check the transaction effective amount defined at Confirmation's response in order to update it at your side.

In some payment methods, such as bank transfer, Customer could deposit a smaller amount (or higher) than the one requested at your side (we cannot enforce Customer's to deposit the exact requested amount). Once we receive the deposit confirmation on our side we will update the amount as the case may be.

After receiving the request P4F API will redirect the Customer to the Merchant's redirection URL according the transaction status (Ok or Not Ok). Merchant should inform the Customer accordingly regarding the transaction outcome, success or fail.

The Merchant's **confirmation URL** will receive the confirmation request with all parameters needed to validate if Pay in was successful or not.

If Pay4Fun receives a non-positive response (any response different from Status 200 OK) from you or no response at all (communication failure), a series of subsequent request retries will be triggered.

Important

Please notice that **you should ONLY respond with a 200 OK response if transaction was received and handled successfully at your side.**

PAY IN CONFIRMATION

Pay4Fun's API will send confirmation requests to your confirmation URL for a 24 hours period, with 10 minutes between requests.

If Pay4Fun's API doesn't receive a Status 200 OK until the last attempt, the retry operation will be terminated and no further attempts will occur.

Important

In case of any communication failure on receiving Pay4Fun's confirmation you should keep listening to your Confirmation URL until receiving the subsequent confirmation and, then, complete the transaction on your side.

The official transaction confirmation will only be made through the informed Merchant's Confirmation URL; the confirmation JSON will be sent through POST to this URL.

The response you will receive at you Confirmation URL looks like the following:

```
{
  "TransactionId": 98765,
  "Amount": 10.00,
  "FeeAmount": 0.00,
  "MerchantInvoiceId": "ABC777",
  "Currency": "USD",
  "Status": "201",
  "LiquidationDate": "2019-01-15",
  "Message": "success",
  "CustomerEmail": "customer@email.com",
  "Sign": "0CE3325211878F6A9131252128EEAA1EB0398B594",
  "PaymentMethod": "Pix"
}
```

RESPONSE PARAMETERS SPECIFICATION

Parameter	Type	Description
TransactionId	String	P4F's transaction unique identifier
Amount	Decimal	Transaction amount
FeeAmount	Decimal	Transaction's fee amount
merchantInvoiceId	String	Merchant's transaction unique identifier
Currency		Pay in currency (ISO 4217 - 3 characters code)
Status	String	Status 201 denotes transaction success. Status 102 denotes PENDING transaction Any other status denotes failed transaction
LiquidationDate	String	Date the transaction's amount will be transferred to Merchant's Available Balance.
Message	String	Success or fail message
CustomerEmail	String	E-mail that has originated the transaction. Use it as additional information on compliance processes
Sign	String	Contains details of the response hashed according Pay in Sign below
PaymentMethod	String	Informs payment method type used by Customer (boleto, PIX, bank transfer, etc.)

The response contains a hashed parameter to provide you way to confirm that information hasn't been tampered: the parameter **Sign**.

PAY IN CONFIRMATION

PAY IN SIGN HASH SEED STRING

The parameter **Sign**, allows you to confirm the veracity of all relevant information contained within the response. The Sign hash should be used to confirm the information has not be tampered.

Considering a Transaction amount of **10.00 USD** related to the merchant's invoice ID **ABC777** and assuming merchant's ID is **34567** and the response was successful (response's Status parameter equals to 201), the seed string for hash generation would be (in that order):

1. merchant's ID;
2. transaction's amount with cents;
3. merchant's invoice ID;
4. response's Status parameter

Seed string = **345671000ABC777201**

Warning

These parameters' seed strings **are different from the hash seed for API Pay in original request**. Please, be aware of that!

Check Appendix 1 – Generating the Hash section for more details.

Pay In Confirmation Checklist

Before going any further and definitely before the Go Live, we suggest you go through this highly recommend checklist to confirm the integration is ready to go and avoid problems in the future:

CHECKLIST

- You have carefully read the documentation until now, specially the yellow and red boxes containing very important information.
- You have your confirmation URL up and running, listening to Pay4Fun's Pay In confirmation requests.
- At your confirmation URL, you are responding Pay In confirmation with 200 OK status **IF, and ONLY IF**, you have successfully processed the transaction at your side.

Pay In Query Status

This API endpoint allows the Merchant to retrieve the status of a list of Merchant's invoice ids.

Important

We recommend sending just **one batch with several Invoice ids** instead of sending several batches, all at the same time, with just one Invoice id each to avoid blockages!

The request header should contain two parameters: **merchantId** and **hash** for security and authorization matters.

PAY IN QUERY STATUS REQUEST:

POST <P4F_API>/1.0/go/transaction/

Header

```
{
  merchantId: 123456,
  hash: "xxxxxxxxxx"
  Content-Type: "application/json"
}
```

Body (list of Merchant's Invoice IDs - Pay In unique identifiers)

```
[
  "ABC777", "ABD778"
]
```

REQUEST PARAMETERS SPECIFICATION

Parameter	Type	Description	Mandatory
merchantId	String	Merchant unique ID	yes
hash	String	Request hash to avoid fraud (check Appendix 1 – Generating the Hash section)	Yes
merchantInvoiceId	String	List of Merchant's Pay In unique identifiers	Yes

Concatenate the following information, in that order, to obtain the seed string for hash generation for Pay in Confirmation requests (check Appendix 1 – Generating the Hash section):

- Merchant's ID
- Merchant's secret

Assuming merchant's Id is **34567** and merchant's secret is "**abcdef**", the seed string for hash generation would be:

Seed string = **34567abcdef**

Important

The Merchant's secret should be kept safe because is the secret factor that prevents others to generate valid hash.

PAY IN QUERY STATUS

PAY IN QUERY STATUS RESPONSE:

Notice that the response will contain information regarding any Merchant Invoice ID present in the request, even if this ID hasn't been found! In those cases, the Status will be **NotFound** which means there's no transaction related to that specific Merchant Invoice ID.

The response for the Pay in Query Status request is like below:

```
[
  {
    "transactionId": 31,
    "merchantInvoiceId": "ABC777",
    "status": "Verified",
    "sign": "878F6A3891313521128EE1EB03984A1A9B94EAE9F240",
    "amount": 222.60,
    "customerEmail": "customer@email.com",
    "liquidationDate": "2019-02-25"
  },
  {
    "transactionId": 0,
    "merchantInvoiceId": "ABD778",
    "status": "NotFound",
    "sign": "878F6A389131E58AE25210CE335A1A9BCB594EAE9F2",
    "amount": 0,
    "customerEmail": null,
    "liquidationDate": null
  }
]
```

RESPONSE PARAMETERS SPECIFICATION

Parameter	Type	Description
transactionId	String	Pay4Fun Transaction ID
merchantInvocield	String	Merchant's Transaction ID
status	String	Pay in transaction status
sign	String	Hash sign to allow veracity confirmation
amount	Decimal	Transaction amount
customerEmail	String	Customer's e-mail
LiquidationDate	String	Date the transaction's amount will be transfer to Merchant's available balance.

Regarding the pay in transaction's statuses you should consider the following:

- **PENDING:** transaction is waiting for Customer's payment;
- **LIQUIDATION:** transaction was successfully processed and the amount will be transferred to Merchant's Available Balance at liquidation date;
- **VERIFIED or RELEASED:** transaction was successfully processed and the amount transferred to Merchant's Available Balance;
- **DECLINED:** transaction was irreversible declined;

The sign parameter contained within the response is built as the following description:

Consider a Transaction amount of **10.00 USD** related to the merchant's invoice ID **ABC777**.

Assuming merchant's ID is **34567** and transaction status is **Verified**, the seed string for hash generation would be (in that order):

1. merchant's ID;
2. merchant's invoice ID ;
3. transaction's amount with cents;
4. transaction's status;

Seed string = **34567ABC7771000Verified**

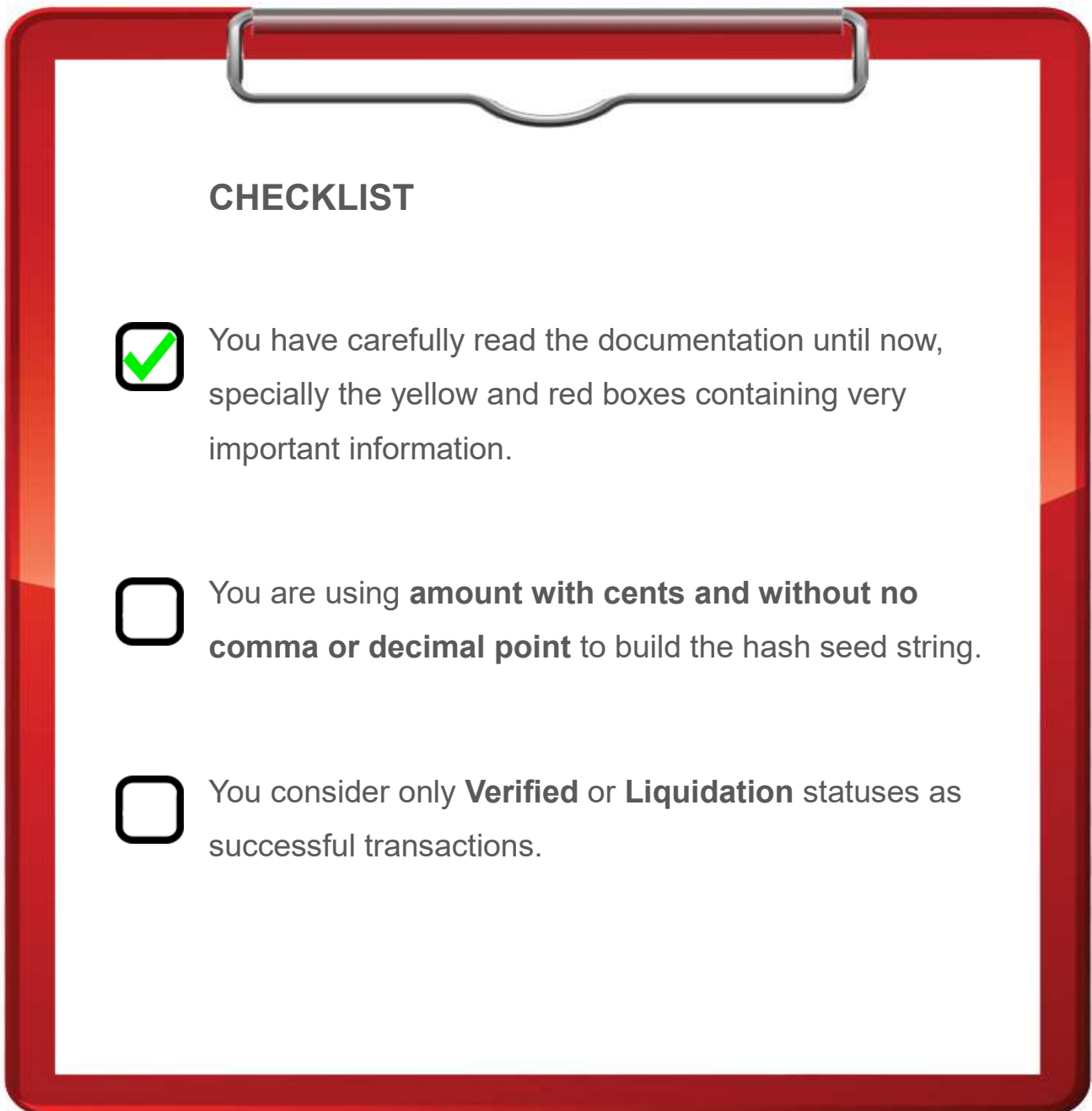
Important

Only Transactions with **Verified** or **Liquidation** statuses should be considered as successfully processed. Any other status indicates unsuccessful a transaction.

Check Appendix 1 – Generating the Hash section for more details.

Pay In Query Status Checklist

Before going any further and definitely before the Go Live, we suggest you go through this highly recommend checklist to confirm the integration is ready to go and avoid problems in the future:



CHECKLIST

- You have carefully read the documentation until now, specially the yellow and red boxes containing very important information.
- You are using **amount with cents and without no comma or decimal point** to build the hash seed string.
- You consider only **Verified** or **Liquidation** statuses as successful transactions.

Payment Methods

Retrieve a list with all available payment methods for P4F GO API.

PAYMENT METHODS API REQUEST:

GET <P4F_API>/1.0/go/PaymentMethods/

Header

```
{
  merchantId: 123456,
  hash: "xxxxxxxxxx"
  Content-Type: "application/json"
}
```

An example of a valid PaymentMethods request is shown below:

```
[{
  "paymentMethodId": 1987,
  "name": "Itau",
  "paymentMethodName": "Bank Transfer"
},
{
  "paymentMethodId": 2009,
  "name": "PIX",
  "paymentMethodName": "Bank Transfer"
}, ...
]
```

PAYMENT METHOD HASH SEED STRING

Concatenate the following information, in that order, to obtain the seed string for hash generation for Pay in requests (check Appendix 1 – Generating the Hash section):

- Merchant's ID
- Merchant's secret

Assuming merchant's ID is **34567** and merchant's secret is "**abcdef**", the seed string for hash generation would be:

Seed string = **34567abcdef**

Appendix 1 – Generating the Hash or Sign

Keeping the communication safe is our top priority, therefore, to avoid any impersonation attempt, all request should contain a **Hash-based Message Authentication Code** (HMAC) by using the SHA256 hash function.

Specific information that are part of the request should be encrypted together with the Merchant's secret to ensure that all information received by P4F is exactly the same sent by the Merchant. You will be using your Merchant Key as encryption key for this purpose.

Keep in mind that each request type demands a particular seed string, formed by fragments of the own request, to generate the hash and any request containing incorrect hash will be declined; the error message you will receive will be:

invalid_security_hash

The **Appendix 1 – Generating the Hash or Sign** contains a guideline on generating hashes coded in C sharp.

Please, for an on-line tool for generating hashes access the URL below:

<https://dotnetfiddle.net/bYQfgP>

Using this online tool you will be able to generate your hashes for testing purposes and confirm you are generating hashes accordingly.

The method below (C Sharp) should be used as a guideline to generate all required API hashes.

The encryption key to be used is your **Merchant Key**.

```
public string HMACSHA256(string toBeEncrypted, string
merchantKey)
{
    byte[] key = Encoding.UTF8.GetBytes(merchantKey);
    using (HMACSHA256 hmac = new HMACSHA256(key))
    {
        hmac.Initialize();
        byte[] bytes_hmac_in =
        Encoding.UTF8.GetBytes(toBeEncrypted);

        byte[] bytes_hmac_out =
        hmac.ComputeHash(bytes_hmac_in);

        string str_hmac_out =
        BitConverter.ToString(bytes_hmac_out);

        str_hmac_out = str_hmac_out.Replace("-", "");

        return str_hmac_out;
    }
}
```


Appendix 2 – Error Messages

Above listed are the possible error messages sent by Pay4Fun in response to your API's request:

ERROR MESSAGES	
Error message	Description
customer_not_authorized	Customer suspended or blocked thus not able to perform transactions
declined_by_customer	Transaction was declined intentionally by the Customer.
external_invoice_id_already_used	There is a previous Transaction using the specified Merchant's Invoice ID.
invalid_customer_main_id	The target customer has a different Main ID that the value specified within the request.
invalid_request	The requested sent by Merchant is not valid.
invalid_request_currency	The Currency sent within the request is not valid.
invalid_security_hash	Something is wrong with the Hash sent within the request. Check the instructions on how generate the Hash correctly.
merchant_account_not_live	The Merchant account is not yet Live.
merchant_ip_not_authorized	The IP used to perform API request is not whitelisted.
merchant_not_authorized	The Merchant's account is not enabled.

APPENDIX 2 – ERROR MESSAGES

merchant_not_found	The Merchant ID sent within the request is not valid.
processing_error	Something very bad has happened while processing Merchant's request.
maximum_amount_exceeded	The maximum allowed transaction amount was exceeded
minimum_amunt_not_reached	The minimum allowed transaction amount was not reached

Appendix 3 – Merchant Labels

Merchant labels are a way to tag transactions (pay in or payout) as originated from a specific merchant's brand or website.

Accessing the merchant's back-office, through the menu **Account / Api / Labels** is possible to manage your labels (create or rename).

Labels have only two properties: ID and Name. The Name has its length limited to 12 characters

In order to label a transaction as originated from the specific brand or website, Merchant should use a valid Label ID to fill up the Api request parameter **LabelId**.

Merchant's reports will show labeled transaction's label in the reports results.

Keep in mind that transactions will only be labeled with a specific label if the request parameter **LabelId** is filled with a valid Label ID otherwise the transaction will remain untagged.

Appendix 4 – Security issues using iframes

The use of iframes is not allowed, because your site would become vulnerable to cross-site attacks.

The use of iframes brings a sort of security risks:

- 1- You may get a submittable malicious web form, phishing your users' personal data.
- 2- A malicious user can run a plug-in.
- 3- A malicious user can change the source site URL.
- 4- A malicious user can hijack your users' clicks.
- 5- A malicious user can hijack your users' keystrokes.
- 6- Put your visitors at risk to the XSS attacks.

For more references, please visit the following:

- <https://www.ostraining.com/blog/webdesign/against-using-iframes/#:~:text=Reason%20%231.&text=If%20you%20create%20an%20iframe,e,change%20the%20source%20site%20URL>.
- https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html#X-Frame-Options_Header_Types

Several articles regarding this issue can be found Googling the keywords "iframe" and "security" if you wish more information.

Appendix 5 – Pay4Fun Logo

You can find all Pay4Fun logos on:

https://blog.p4f.com/wp-content/uploads/2021/03/Package_P4F.zip