



---

# Payments API Documentation

---

**Version** 1.0.8

**Date** 23/12/2024

# Table of Contents

<b>Changelog</b>	3
<b>Definitions</b>	4
<b>Format Definitions</b>	6
Examples	7
<b>Prerequisites</b>	8
<b>Parameter requirements</b>	8
<b>Responses</b>	9
Web-redirect	9
JSON	9
<b>Process Flow</b>	10
<b>Configuration</b>	11
<b>Parameters</b>	12
HEADERS	12
BODY	13
<b>Select Card Type Pages</b>	22
FAC	22
FGB	23
<b>Testing - Credit/Debit Card</b>	24
FAC	24
Non-AVS Hosted Page	23
AVS Hosted Page	23
Test Cards	24
Additional Notes	26
FGB	27
Non-AVS Hosted Page	27
AVS Hosted Page	27
Test Cards	27

Additional Notes	27
Example Code	29
HTML	29
PHP	30
JavaScript	31
<b>Transaction Response</b>	32
Response Parameters	32
Example	36
FAC	37
FGB	37
<b>Transaction Fee Rates</b>	39
<b>FAQs</b>	40
What is the difference between the new and old APIs?	40
Request Parameters	41
Response Parameters	41
How to get the money into my Bank Account?	43
What is an API Key?	45
How to get my API Key?	46
How do I generate a new API Key?	48
What is the API Key of the TEST Account?	49
<b>Appendices</b>	50
Appendix 1 - Other FGB pages	50
Success transaction summary	50
Fail transaction summary	51
3DS Authentication password dialog	52

# Changelog

Version	Description	Date
1.0.0	Initial version.	17/06/2021
1.0.1	<ol style="list-style-type: none"> <li>1. Added the <b>Definitions</b> section.</li> <li>2. Added various grammatical improvements..</li> </ol>	18/06/2021
1.0.2	<ol style="list-style-type: none"> <li>1. Updated JM USD Hosted Page Service provider.</li> <li>2. Updated name of document to be clearer.</li> </ol>	23/06/2021
1.0.3	<ol style="list-style-type: none"> <li>1. Updated the formatting of various texts.</li> <li>2. Added the <b>Example Code</b> section.</li> </ol>	29/06/2021
1.0.4	<ol style="list-style-type: none"> <li>1. Updated upgrades to the API request parameters.</li> <li>2. Added new OPTIONAL parameters.</li> <li>3. Revised the Example code section.</li> <li>4. Updated upgrades to the API's JSON response.</li> <li>5. Updated FAC Hosted Page images.</li> </ol>	20/07/2021
1.0.5	<ol style="list-style-type: none"> <li>1. Updated upgrades to the API parameters.</li> <li>2. Revised font for parameter Formats.</li> <li>3. Revised various wording and formatting.</li> <li>4. Added the <b>Select Card Type Pages</b> section.</li> <li>5. Updated non-AVS FAC Hosted Page image.</li> </ol>	10/08/2021
1.0.6	<ol style="list-style-type: none"> <li>1. Updated information regarding changes to BB platform.</li> <li>2. Updated the "data" request parameter to OPTIONAL.</li> <li>3. Revised various wording and formatting.</li> </ol>	01/09/2021
1.0.7	<ol style="list-style-type: none"> <li>1. Revised various wording, images and formatting.</li> <li>2. Updated the language to indicate that both Credit and Debit cards are supported.</li> <li>3. Added "Additional Notes" subsection to both the FAC and FGB sections.</li> <li>4. Updated the "hash" response parameter description language.</li> </ol>	07/10/2021
1.0.8	<ol style="list-style-type: none"> <li>1. Updated API Parameters</li> <li>2. Restored missing BODY Parameter.</li> <li>3. Revised response parameters comparison.</li> </ol>	23/12/2024

# Definitions

- 3DS** Abbreviation for “[3-D Secure](#)”. A protocol designed to be an additional security layer for online credit and debit card transactions.
- API** Abbreviation for “[Application Programming Interface](#)”. A type of software interface, offering a service to other pieces of software.
- API Key** Refer to [What is an API Key](#).
- AVS** Abbreviation for “[Address Verification Service](#)”. A service provided by major card processors to enable merchants to authenticate ownership of a credit or debit card used by a customer.
- CVV or CVV2** Abbreviation for “[Card Verification Value](#)”. A security feature for card not present transactions, where a personal identification number (PIN) cannot be manually entered by the cardholder (as they would during point-of-sale or card present transactions).
- FAC** Abbreviation for “First Atlantic Commerce”. In the context of this document, it is a Hosted Page Service.
- FGB** Abbreviation for “First Global Bank”. In the context of this document, it is a Hosted Page Service.
- hash** By definition, a “[hash](#)” is a cryptographic function which acts on a piece of data of arbitrary size, converting it into another piece of data of fixed size. In the context of this document, it is used for information security and authentication.
- HTTP** Abbreviation for “[Hypertext Transfer Protocol](#)”. It is a standard protocol used for transmitting and communicating data across the world wide web.
- HTTP Status Code** Also known as “[HTTP Response Codes](#)”, this indicates whether a specific HTTP request has been successfully completed.
- JSON** Abbreviation for “[JavaScript Object Notation](#)”. It is a language-independent data format for data interchange, used most commonly by web applications to communicate with a server.
- MD5** Related to “hash”, MD5 is a hashing algorithm (i.e. cryptographic function).
- parse** Refers to the act of “[parsing](#)”. It is the process of analyzing a string of symbols into its constituents to garner greater contextual or applicative value.
- querystring** A [query string](#) is a part of a uniform resource locator (URL) that assigns values to specified parameters.

**sandbox** In the context of this document, “sandbox” can be considered synonymous to “test” or “testing environment”.

**Web-redirects** Also known as “[HTTP redirection](#)”, it is a technique to give more than one URL address to a page, a form, or a whole Web site/application.

# Format Definitions

The parameters passed to the API are all validated against a specific format. This section describes what the format descriptions mean for any given parameter as defined in this document.

<p><b>a</b></p>	<p>Alphabetic.</p> <ul style="list-style-type: none"> <li>● a-z (lowercase)</li> <li>● A-Z (uppercase)</li> <li>● “ ”</li> </ul> <p>Characters with accents are <b>not</b> supported. The “ ” is acceptable only in certain parameters.</p>
<p><b>b</b></p>	<p>Boolean.</p> <ul style="list-style-type: none"> <li>● true: true, 1, “true”, “1”</li> <li>● false: false, 0, “false”, “0”</li> </ul> <p>Booleans will always have a length of 1 (implied).</p>
<p><b>d</b></p>	<p>Dashes.</p> <ul style="list-style-type: none"> <li>● “_”</li> <li>● “ - ”</li> </ul>
<p><b>n</b></p>	<p>Numeric.</p> <ul style="list-style-type: none"> <li>● 0-9</li> <li>● “.”</li> </ul> <p>The “.” is acceptable only in certain parameters.</p>
<p><b>s</b></p>	<p>Special.</p> <ul style="list-style-type: none"> <li>● !#\$%&amp;“ ’ * { } + , / : ; &lt; &gt; = ? @ [ \ ] ^ ` (   ) ~</li> </ul> <p>A subset of these characters are acceptable under certain circumstances for certain parameters.</p>
<p><b>v</b></p>	<p>Variable.</p> <p>This means that any character is permissible, and/or any length is permissible.</p>
<p><b>[ ]</b></p>	<p>Length, or Size.</p> <ul style="list-style-type: none"> <li>● <b>[NUM]</b> <ul style="list-style-type: none"> <li>○ The given parameter may only have NUM length.</li> </ul> </li> <li>● <b>[MIN-MAX]</b> <ul style="list-style-type: none"> <li>○ The given parameter may have any length from MIN (inclusive) to MAX (inclusive).</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• <b>[NUM1   NUM2]</b> <ul style="list-style-type: none"> <li>○ The given parameter may have either NUM1 length only, or NUM2 length only.</li> </ul> </li> </ul>
--	--

These formats (except Boolean, Variable and Length) may be uniquely chained together to form a compound format specification. For example, “**a[v]**” is defined as any alphabetic data with variable length. However, it can be compounded as “**an[v]**”, which would now be defined as alpha-numeric data with variable length.

**Examples**

Here are some examples that may help in reading various format specifications.

<b>an[5]</b>	Alphabetic + Numeric. Commonly referred to as “alphanumeric”. Which results in a-z, A-Z, 0-9 and “.” as valid characters. The “[5]” indicates that this “alphanumeric” data can only be 5 characters long.
<b>v[2-10]</b>	Variable data that can have a minimum of 2 characters, or a maximum of 10 characters.
<b>ans[5   10-15]</b>	Alphabetic + Numeric + Special. Valid characters are a-z, A-Z, 0-9, “.” and any of “!#\$%&''*{}+,:;<>=?@[\\]^`{ }~“. The data can have a length of 5, or a minimum length of 10 to a maximum length of 15.

# WiPay Plugins Payment Request

The official WiPay API endpoint for requesting a Transaction Gateway (Secure Hosted Page).

## Prerequisites

To use this API for LIVE transactions:

1. You must have a [WiPay Business Account](#).
2. Your WiPay Business Account must be [Verified](#).
3. You must have an [API Key](#).

There are no other special requirements for using the API for SANDBOX transactions (testing). However, do note that the [API Key](#) for the Test WiPay Account is **123**.

## Parameter requirements

Requirement	Description
<b>REQUIRED</b>	The <code>parameter</code> must be submitted with the API request. Failure to do so will result in an error response; usually 400-class responses.
<b>OPTIONAL</b>	The <code>parameter</code> may be submitted with the API request. Failure to do so will not result in an error response.

# Responses

## Web-redirect

If the API is configured for Web-redirects, then:

- For **Success** responses, users will be automatically redirected to the Payment Gateway Secure Hosted Page.
- For **Error** responses, users will be automatically redirected to the Response URL (the `response_url` parameter) with the appropriate [response parameters](#) appended in the Response URL as a querystring.

## JSON

The following JSON is returned, if the API is configured for JSON-responses. Please note that the `transaction_id` may not always be present. The API will always attempt to return the `transaction_id` where possible.

- For **Success** responses, HTTP status code 200-class or 300-class responses will be given for successful API requests.

```
{
  "url": "<_UNIQUE_HOSTED_PAGE_PAYMENT_URL_>",
  "message": "<_HTTP_STATUS_MESSAGE_>",
  "transaction_id": "<_TRANSACTION_ID_>"
}
```

- For **Error** responses, HTTP status code 400-class or 500-class responses will be given for unsuccessful API requests.

```
{
  "url": "<_RETURN_URL_WITH_ERROR_IN_QUERY_STRING_PARAMETERS_>",
  "message": "<_ERROR_MESSAGE_DESCRIPTION_>",
  "transaction_id": "<_TRANSACTION_ID_>"
}
```

# Process Flow

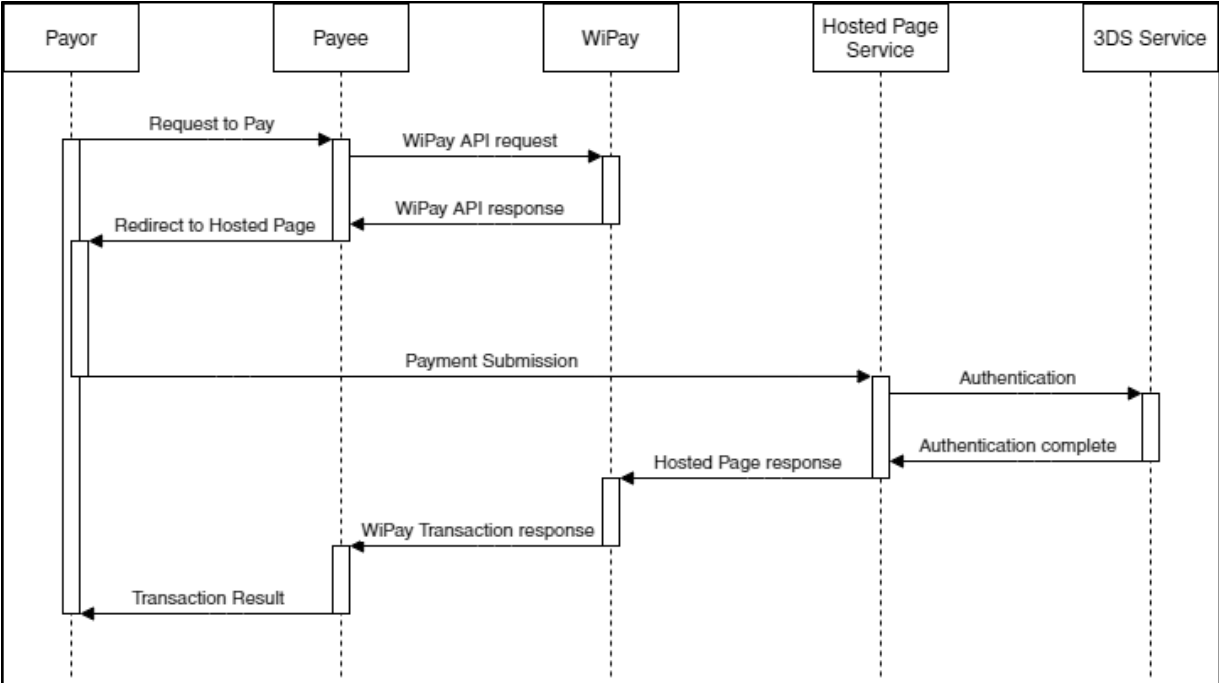
When using this API, the general high-level flow for every transaction is:

1. WiPay Merchant’s web-based application uses this API to obtain a Hosted Page URL.
2. WiPay Merchant’s web-based application uses the received Hosted Page URL to redirect the Customer to the Hosted Page.
3. Customer makes payment using the Hosted Page by submitting their payment credentials.
4. Hosted Page web-redirects to this API’s configured `response_url` with the result of the transaction appended as a query string, containing the [response parameters](#).

The transaction process will always communicate between these entities:

1. The Customer’s web-browser (Payor)
2. The WiPay Merchant’s website (Payee)
3. WiPay’s platform (WiPay)
4. Hosted Page Service
5. 3DS Authentication Service

The following Sequence Diagram shows the communication between all the entities involved for any given transaction.



# Configuration

We strongly recommend that you use the API URL that is most relevant to the country where your WiPay Account is Verified. While these API URLs can function normally across different countries, sending the API request to the native country of the WiPay Account would result in faster responses.

<b>API URL</b>	<a href="https://bb.wipayfinancial.com/plugins/payments/request">https://bb.wipayfinancial.com/plugins/payments/request</a>
	<a href="https://gy.wipayfinancial.com/plugins/payments/request">https://gy.wipayfinancial.com/plugins/payments/request</a>
	<a href="https://jm.wipayfinancial.com/plugins/payments/request">https://jm.wipayfinancial.com/plugins/payments/request</a>
	<a href="https://tt.wipayfinancial.com/plugins/payments/request">https://tt.wipayfinancial.com/plugins/payments/request</a>
<b>HTTP Method</b>	POST

# Parameters

## HEADERS

<b>Accept</b>		<b>OPTIONAL</b>
<b>Example</b>	application/json	
<b>Description</b>	You may explicitly set this request-header field to request either a JSON response, or a web-redirect response from our API.	
<b>Format</b>	as[ v ] See Options.	
<b>Options</b>	application/json, */*	
<b>Notes</b>	<ol style="list-style-type: none"><li>1. Use <code>application/json</code> to get a JSON response.</li><li>2. Use <code>*/*</code>, or omit this request-header field, to get a web-redirect response.</li></ol>	
<b>Content-Type</b>		<b>OPTIONAL</b>
<b>Example</b>	application/x-www-form-urlencoded	
<b>Description</b>	This request-header field must be set so that the parameters defined in request-body can be interpreted.	
<b>Format</b>	as[ v ]	
<b>Options</b>	-	
<b>Notes</b>	-	

## BODY

<b>account_number</b>		<b>REQUIRED</b>
<b>Example</b>	1234567890	
<b>Description</b>	Your LIVE WiPay Account Number.	
<b>Format</b>	n[10]	
<b>Options</b>	-	
<b>Notes</b>	<ol style="list-style-type: none"> <li>If <b>environment</b> is <b>sandbox</b>, then you must use the WiPay SANDBOX Account Number <b>1234567890</b>.</li> </ol>	
<b>avs</b>		<b>OPTIONAL</b>
<b>Example</b>	0	
<b>Description</b>	This enables AVS on the payment gateway. This also enables the <b>AVS-only</b> parameters that can be sent to pre-fill the fields on the AVS form.	
<b>Format</b>	b	
<b>Options</b>	0, 1	
<b>Notes</b>	<ol style="list-style-type: none"> <li><b>AVS-only</b> parameters are meant to be filled out by the Payor.</li> <li><b>AVS-only</b> parameters' information are meant to be consistent with the KYC submitted to the Payor card's issuing Bank.</li> </ol>	
<b>card_type</b>		<b>OPTIONAL</b>
<b>Example</b>	mastercard	
<b>Description</b>	The payment processing network of the Payor's card.	
<b>Format</b>	a[v] See Options.	
<b>Options</b>	mastercard, visa	
<b>Notes</b>	<ol style="list-style-type: none"> <li>If this parameter is not provided in the API request, Payors will first encounter a <a href="#">Select Card Type page</a> before entering their Card</li> </ol>	

	information.	
<b>country_code</b>		<b>REQUIRED</b>
<b>Example</b>	TT	
<b>Description</b>	The country in which your <b>account_number</b> is registered.	
<b>Format</b>	a[2] <a href="#">ISO 3166-1 alpha-2</a> . See Options.	
<b>Options</b>	BB, JM, TT	
<b>Notes</b>	-	
<b>currency</b>		<b>REQUIRED</b>
<b>Example</b>	TTD	
<b>Description</b>	The currency of the <b>total</b> for this transaction.	
<b>Format</b>	a[3] <a href="#">ISO 4217 alpha code</a> . See Options.	
<b>Options</b>	JMD, TTD, USD	
<b>Notes</b>	1. This depends on <b>country_code</b> , since supported currencies vary based on the country.	
<b>environment</b>		<b>REQUIRED</b>
<b>Example</b>	sandbox	
<b>Description</b>	Determines if the payment gateway will be configured to the test environment, or not.	
<b>Format</b>	a[v] See Options.	
<b>Options</b>	live, sandbox	
<b>Notes</b>	-	

<b>fee_structure</b>		<b>REQUIRED</b>
<b>Example</b>	customer_pay	
<b>Description</b>	Controls how and who pays the WiPay Transaction Fee.	
<b>Format</b>	ad[v] See Options.	
<b>Options</b>	customer_pay, merchant_absorb, split	
<b>Notes</b>	1. This directly affects the final total of the transaction.	
<b>method</b>		<b>REQUIRED</b>
<b>Example</b>	credit_card	
<b>Description</b>	Determines the payment method your customers will use to pay you.	
<b>Format</b>	ad[v] See Options.	
<b>Options</b>	credit_card	
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. This parameter depends on country_code and currency. <ol style="list-style-type: none"> <li>a. For example, voucher is available for country_code of TT and currency of TTD only (voucher option <u>coming soon</u>).</li> </ol> </li> <li>2. The credit_card option supports both Credit and Debit cards; this support is primarily determined by the card's Issuing Bank.</li> </ol>	
<b>order_id</b>		<b>REQUIRED</b>
<b>Example</b>	oid_123-aBc	
<b>Description</b>	Your application's custom unique identifier for this transaction.	
<b>Format</b>	ad[1-16 1-48] See Notes.	
<b>Options</b>	-	

<b>Notes</b>	<ol style="list-style-type: none"> <li>1. Must begin and end with an alphanumeric character.</li> <li>2. Up to 48 characters for <a href="#">FAC</a>.</li> <li>3. Up to 16 characters for <a href="#">FGB</a>.</li> <li>4. This parameter depends on <a href="#">country_code</a> and <a href="#">currency</a>.</li> <li>5. This parameter should always be <a href="#">unique</a>.</li> </ol>	
<b>origin</b>		<b>REQUIRED</b>
<b>Example</b>	WiPay-example_app	
<b>Description</b>	Your application's custom unique identifier for this transaction.	
<b>Format</b>	ad[1-32]	
<b>Options</b>	-	
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. Must begin and end with an alphanumeric character.</li> </ol>	
<b>response_url</b>		<b>REQUIRED</b>
<b>Example</b>	https://tt.wipayfinancial.com/response/	
<b>Description</b>	Your application's designated URL to handle WiPay's transaction response.	
<b>Format</b>	as[1-255] <a href="#">Generic URI syntax</a> .	
<b>Options</b>	-	
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. This parameter will be appended with a querystring of <a href="#">response parameters</a>.</li> </ol>	
<b>total</b>		<b>REQUIRED</b>
<b>Example</b>	10.00	
<b>Description</b>	The price of your product or service.	
<b>Format</b>	n See Notes.	

<b>Options</b>	-
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. This value must be correct to 2 decimal places.</li> <li>2. Based on the configured <b>fee_structure</b> for the payment request, payors may see a final <b>total</b> affected by your WiPay account's transaction rates. Please see the <a href="#">Transaction Fee Rates</a> section for further details.</li> <li>3. The minimum value for this parameter is <b>\$1.00 USD</b> or the <b>currency</b>-value equivalent.</li> </ol>
<b>version</b>	<b>OPTIONAL</b>
<b>Example</b>	1.0.0
<b>Description</b>	Your application's official version number.
<b>Format</b>	ans[1-16] <a href="#">Semantic Versioning</a> (recommended)
<b>Options</b>	-
<b>Notes</b>	-
<b>addr1</b>	<b>OPTIONAL</b>
<b>Example</b>	-
<b>Description</b>	The Street Address of the Payor's residence
<b>Format</b>	adn[1-50]
<b>Options</b>	-
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. <b>AVS-only</b> pre-fill is supported.</li> </ol>
<b>addr2</b>	<b>OPTIONAL</b>
<b>Example</b>	-
<b>Description</b>	The Apartment, Suite, Floor etc. of the Payor.

<b>Format</b>	adn [0-50]	
<b>Options</b>	-	
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. AVS-only pre-fill is supported.</li> <li>2. This parameter can be empty.</li> </ol>	
<b>city</b>		<b>OPTIONAL</b>
<b>Example</b>	-	
<b>Description</b>	The city of residence of the Payor.	
<b>Format</b>	a [1-30]	
<b>Options</b>	-	
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. AVS-only pre-fill is supported.</li> </ol>	
<b>country</b>		<b>OPTIONAL</b>
<b>Example</b>	-	
<b>Description</b>	The country in which the Payor legally resides.	
<b>Format</b>	a [2] <a href="#">ISO 3166-1 Alpha 2</a> .	
<b>Options</b>	-	
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. AVS-only pre-fill is supported.</li> </ol>	
<b>email</b>		<b>OPTIONAL</b>
<b>Example</b>	-	
<b>Description</b>	The Payor's contact email.	
<b>Format</b>	ans [1-50] <a href="#">RFC 822</a> .	

<b>Options</b>	-
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. <b>AVS-only</b> pre-fill is supported.</li> <li>2. This will enable the Payor to receive an automated email upon Transaction submission (both success and fail).</li> <li>3. If present, this parameter is appended to the <a href="#">response parameters</a> as <b>customer_email</b>.</li> </ol>
<b>fname</b>	<b>OPTIONAL</b>
<b>Example</b>	-
<b>Description</b>	The Payor's legally registered first name.
<b>Format</b>	adn[1-30]
<b>Options</b>	-
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. <b>AVS-only</b> pre-fill is supported.</li> <li>2. Both <b>fname</b> and <b>lname</b> parameters are concatenated and appended to the <a href="#">response parameters</a> as <b>customer_name</b> for AVS transactions.</li> </ol>
<b>lname</b>	<b>OPTIONAL</b>
<b>Example</b>	-
<b>Description</b>	The Payor's legally registered last name.
<b>Format</b>	adn[0-30]
<b>Options</b>	-
<b>Notes</b>	<ol style="list-style-type: none"> <li>1. <b>AVS-only</b> pre-fill is supported.</li> <li>2. This parameter can be empty.</li> <li>3. If provided, this parameter <u>must</u> be used together with <b>fname</b>.</li> </ol>
<b>name</b>	<b>OPTIONAL</b>
<b>Example</b>	-
<b>Description</b>	The Payor's legally registered full name.

<b>Format</b>	adn[1-60]	
<b>Options</b>	-	
<b>Notes</b>	<ol style="list-style-type: none"> <li>AVS-only pre-fill is supported. <ol style="list-style-type: none"> <li>For AVS transactions, if this parameter is provided and <b>fname</b> and <b>lname</b> were not provided, then this parameter will always attempt to split into an <b>fname</b> and <b>lname</b> pair.</li> <li>The <b>lname</b> will always be parsed as the last word.</li> </ol> </li> <li>If present, this parameter is appended to the <a href="#">response parameters</a> as <b>customer_name</b> for non-AVS transactions.</li> </ol>	
<b>phone</b>		<b>OPTIONAL</b>
<b>Example</b>	-	
<b>Description</b>	The Payor's contact phone number.	
<b>Format</b>	ns[1-20] <a href="#">E.164</a> (recommended). See Notes.	
<b>Options</b>	-	
<b>Notes</b>	<ol style="list-style-type: none"> <li>AVS-only pre-fill is supported. <ol style="list-style-type: none"> <li>This parameter is validated using <a href="#">Google's libphonenumber library</a>.</li> </ol> </li> <li>If present, this parameter is appended to the <a href="#">response parameters</a> as <b>customer_phone</b> for non-AVS transactions.</li> </ol>	
<b>state</b>		<b>OPTIONAL</b>
<b>Example</b>	-	
<b>Description</b>	The US state in which the Payor resides.	
<b>Format</b>	a[2] <a href="#">ISO 3166-2:US</a> (second part)	
<b>Options</b>	-	
<b>Notes</b>	<ol style="list-style-type: none"> <li>AVS-only pre-fill is supported.</li> <li>This applies only to US-based residency; i.e. if <b>country</b> is <b>US</b>.</li> </ol>	

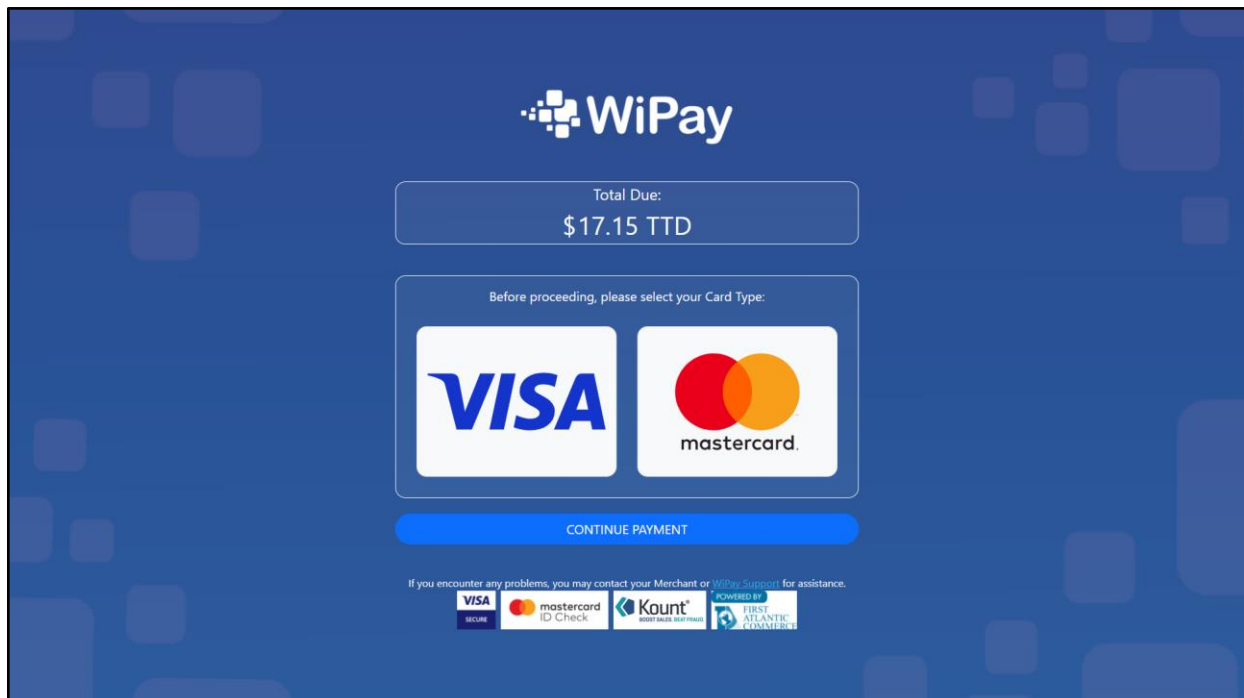
<b>zipcode</b>		<b>OPTIONAL</b>
<b>Example</b>	-	
<b>Description</b>	The Zip/Postal Code of the Payor.	
<b>Format</b>	an[1-10]	
<b>Options</b>	-	
<b>Notes</b>	1. AVS-only pre-fill is supported.	

# Select Card Type Pages

The Select Card Type page is where Payors select the payment processing network of their Card. Payors may see this page before entering their card information.

This is a mandatory step in the payment process. Payors will only see this page if the `card_type` parameter is not present in the original API request.

## FAC



# FGB

Version: 21.3.0-302

Build time: 2021-06-09T17:21:46+0200

Last commit: 420a694



Your Commercial Bank from Graceland

Amount \$ 17.15 JMD

First Global hosted payment page

Please select payment method



Cancel

Continue

# Testing - Credit/Debit Card

WiPay considers testing to be the single most important aspect of any integration. It is to ensure that your web-based application behaves consistently and predictably when it uses our external service (API).

Generally, this API behaves consistently for both LIVE and SANDBOX transactions - barring any specific response parameter differences as would be expected for unique transaction attempts.

Please note:

- When conducting test transactions (SANDBOX i.e. **environment** is set to **sandbox**), all reporting features are disabled;
  - Payees and Payors do not receive automated emails.
  - Transaction History information will not be available.

## Test Cards

You may use these cards to test for different types of responses from the FAC Hosted Page.

Please note that:

- Any expiry date and any 3 digit CVV2 value will work for these test cards.
- All card numbers not listed above are defaulted to Normal Approval.
- “Normal Approval” means ResponseCode=1, ReasonCode=1.
- “Normal Decline” means ResponseCode=2, ReasonCode=2

<b>Mastercard</b>	5111111111111111	Normal Approval, CVV2Result=M
	51111111111112222	Normal Approval, CVV2Result=N
	5333333333332222	Normal Approval, CVV2Result=U
	5444444444442222	Normal Approval, CVV2Result=P
	5555555555552222	Normal Approval, CVV2Result=S
	5555666666662222	Normal Decline, OriginalResponseCode=05, CVV2Result=N
	51111111111113333	Normal Decline, OriginalResponseCode=05

	5111111111114444	Normal Approval, AVSResult=Y
	5111111111115555	Normal Approval, AVSResult=A
	5111111111116666	Normal Approval, CVV2Result=M, AVSResult=Z
	5111111111117777	Normal Approval, CVV2Result=M, AVSResult=N
	5111111111118888	Normal Approval, CVV2Result=N, AVSResult=U
	5111111111119999	Normal Decline, OriginalResponseCode=98
	5111111111110000	Normal Decline, OriginalResponseCode=91
	5222222222222222	Normal Approval, CVV2Result=N, AVSResult=U
<b>VISA</b>	4111111111111111	Normal Approval, CVV2Result=M
	4111111111112222	Normal Approval, CVV2Result=N
	4333333333332222	Normal Approval, CVV2Result=U
	4444444444442222	Normal Approval, CVV2Result=P
	4555555555552222	Normal Approval, CVV2Result=S
	4666666666662222	Normal Decline, OriginalResponseCode=05, CVV2Result=N
	4111111111113333	Normal Decline, OriginalResponseCode=05
	4111111111114444	Normal Approval, AVSResult=M
	4111111111115555	Normal Approval, AVSResult=A
	4111111111116666	Normal Approval, AVSResult=Z
	4111111111117777	Normal Approval, AVSResult=N
	4111111111118888	Normal Approval, AVSResult=G

	4111111111119999	Normal Decline, OriginalResponseCode=98
	4111111111110000	Normal Decline, OriginalResponseCode=91
	4222222222222222	Normal Approval, CVV2Result=M, AVSResult=N


### Additional Notes

1. The Payor's Card may be debited in TTD despite the API configuration; for example, if the API request was set to **currency** as **USD**. This would occur if your Card was issued locally in Trinidad and Tobago. In these cases, the Payee is also credited the transactional value equivalent in TTD.
2. The Payor's Card may be debited in USD despite the API configuration; for example, if the API request was set to **currency** as **TTD**. This would occur if your Card was NOT issued locally in Trinidad and Tobago. In these cases, the Payee is also credited the transactional value equivalent in USD.
3. In the above cases, WiPay uses a static conversion rate of USD to TTD of **6.80**.

# FGB

## Non-AVS Hosted Page

Version: 21.3.0-302      Build time: 2021-06-09T17:21:46+0200      Last commit: 420a694



**Amount \$ 10.00 JMD** First Global hosted payment page

Cardholder Name

Card Number \*

Expiration Month \*

Expiration Year \*

Card Code \*   
(last three digits on the back of your card)

\* **Mandatory fields**

## AVS Hosted Page

Coming soon.

## Test Cards

You may use these cards to test for different types of responses from the FGB Hosted Page.

Please note that:

- Any value for “Cardholder Name” on the Hosted Page may be used (may even remain empty).
- Any 2-digit “Expiration Month”, “Expiration Year”, and 3-digit “Card Code” will work for these test cards.
- The provided cards guarantee an APPROVED (success) response.
- Any other cards will result in a FAILED or DECLINED (fail) response.

- When using a provided card, if prompted for password authentication, please use the same text given for the “Personal Message” (see [Appendix 1 - 3DS Authentication Password dialog](#)).
- The FGB Hosted Page will always show the Payor an intermediary transaction result summary page before redirecting away.
- Redirecting away from the FGB Hosted Page occurs only when the Payor clicks the “Return to Shop” button at the transaction result summary page (See the transaction summary pages in [Appendix 1](#)).

<b>Mastercard</b>	5210000010001001	3DS, 3DSResCode=1, AuthStatus=Y
	5204740000002711	FF3DS2+3DSM, 3DSResCode=1, AuthStatus=Y
<b>VISA</b>	4035874000424977	3DS, 3DSResCode=1, AuthStatus=Y
	4265880000000007	FF3DS2+3DSM, 3DSResCode=1, AuthStatus=Y

### Additional Notes

1. -

## Example Code

To serve as a quick starting point, this section provides working example implementations of the WiPay Plugins Payment Request API using the minimum set of parameters.

You may use any modern, popular Web API-capable library of your choice. The recommendations given in this section are based on our users' feedback, which we know have been tried, tested and proven working.

We do not recommend using the provided example code as-is for production environments.

## HTML

The Old API was popularly implemented as an HTML form. This approach is also possible with the New API. Upon clicking the "Checkout" button, this will always automatically redirect the Payor to the Hosted Page (no JSON response).

```
<form action="https://tt.wipayfinancial.com/plugins/payments/request"
method="POST">
  <input type="hidden" name="account_number" value="1234567890">
  <input type="hidden" name="avs" value="0">
  <input type="hidden" name="country_code" value="TT">
  <input type="hidden" name="currency" value="TTD">
  <input type="hidden" name="data" value="{&quot;a&quot;:&quot;b&quot;}">
  <input type="hidden" name="environment" value="sandbox">
  <input type="hidden" name="fee_structure" value="customer_pay">
  <input type="hidden" name="method" value="credit_card">
  <input type="hidden" name="order_id" value="oid_123-aBc">
  <input type="hidden" name="origin" value="WiPay-example_app">
  <input type="hidden" name="response_url"
value="https://tt.wipayfinancial.com/response/">
  <input type="hidden" name="total" value="10.00">
  <!-- Redirect occurs after clicking Checkout -->
  <input type="submit" value="Checkout">
</form>
```

## PHP

We recommend using PHP's cURL library. This is used for implementing the API on the server-side.

```
$curl =
curl_init('https://tt.wipayfinancial.com/plugins/payments/request');
curl_setopt_array($curl, [
    CURLOPT_FOLLOWLOCATION => false,
    CURLOPT_HEADER => false,
    CURLOPT_HTTPHEADER => [
        'Accept: application/json',
        'Content-Type: application/x-www-form-urlencoded'
    ],
    CURLOPT_POST => true,
    CURLOPT_POSTFIELDS => http_build_query([
        'account_number' => '1234567890',
        'avs' => '0',
        'country_code' => 'TT',
        'currency' => 'TTD',
        'data' => '{"a":"b"}',
        'environment' => 'sandbox',
        'fee_structure' => 'customer_pay',
        'method' => 'credit_card',
        'order_id' => 'oid_123-aBc',
        'origin' => 'WiPay-example_app',
        'response_url' => 'https://tt.wipayfinancial.com/response/',
        'total' => '10.00'
    ]),
    CURLOPT_RETURNTRANSFER => true
]);
$result = curl_exec($curl);
curl_close($curl);
# result in JSON format (header)
$result = json_decode($result);
# perform redirect
header("Location: {$result->url}");
die();
```

## JavaScript

We recommend using jQuery's \$.ajax() where possible, however modern vanilla Javascript is more than capable. This is used for implementing the API on the client-side.

```
var headers = new Headers();
headers.append('Accept', 'application/json');

var parameters = new URLSearchParams();
parameters.append('account_number', '1234567890');
parameters.append('avs', '0');
parameters.append('country_code', 'TT');
parameters.append('currency', 'TTD');
parameters.append('data', '{"a":"b"}');
parameters.append('environment', 'sandbox');
parameters.append('fee_structure', 'customer_pay');
parameters.append('method', 'credit_card');
parameters.append('order_id', 'oid_123-aBc');
parameters.append('origin', 'WiPay-example_app');
parameters.append('response_url',
'https://tt.wipayfinancial.com/response/');
parameters.append('total', '10.00');

var options = {
  method: 'POST',
  headers: headers,
  body: parameters,
  redirect: 'follow'
};

fetch('https://tt.wipayfinancial.com/plugins/payments/request', options)
  .then(response => response.text())
  .then(result => {
    // result in JSON format (header)
    result = JSON.parse(result);
    // perform redirect
    window.location.href = result.url;
  })
  .catch(error => console.log('error', error));
```

# Transaction Response

Transaction responses always occur as a web-redirect from the Hosted Payment Page to the `response_url` submitted in the original API request to the WiPay Plugins Payment Request API.

After the payor enters their payment credentials and/or information, a web-redirect occurs to the `response_url`. This is a GET request to the `response_url`. The `response_url` is appended with a URL-encoded querystring containing the result and data of the Transaction that just occurred - commonly referred to as the “response parameters.”

When implementing this API, it is intended that the result of the Transaction is parsed from the querystring in the `response_url`. Thus, we strongly recommend that the `response_url` is a dedicated endpoint on your web-based application’s domain. This will allow you to implement your own custom logic to parse the response parameters and handle the rest of your application’s logic concerning the transaction in one place.

## Response Parameters

The following table defines all the possible response parameters. For some response parameters, their specific formats may vary between Hosted Page Services (FAC/FGB). Additionally, some response parameters may be **conditionally absent**.

<code>card</code>	<p>The padded card number used by the Payor for the Transaction. <b>Only</b> the <u>last FOUR (4) digits</u> of the Payor’s card will be exposed.</p> <p><b>FAC (example):</b> XXXXXXXXXXXX1111</p> <p><b>FGB (example):</b> (VISA) . . . 0026</p>
<code>currency</code>	<p>The currency of the Transaction’s <b>total</b>.</p> <p>This will always be the same as the <code>currency</code> request parameter sent in the original API request.</p>
<code>customer_address</code>	<p><b>conditionally absent</b></p> <p>The full legally registered address of the Payor (Card Holder).</p> <p>This parameter is always composed in the following format: <i>addr1, addr2, city, state zipcode, country</i></p>

	<p>Each of these component strings may be pre-filled in the AVS-enabled API request, or, will be defined when the Payor completes an AVS-enabled Transaction.</p>
customer_company	<p><b>conditionally absent</b></p> <p>The legally registered name of the Payor's (Card Holder's) company.</p> <p>This parameter is application specific; it is supported only by select <b>origins</b>.</p> <p>Please <a href="#">Contact Us</a> to coordinate for supporting this parameter in your application.</p>
customer_email	<p><b>conditionally absent</b></p> <p>The contact email for the Payor (Card Holder).</p> <p>This parameter may be pre-filled in the AVS-enabled API request, or, will be defined when the Payor completes an AVS-enabled Transaction.</p> <p>If present in the original request, this parameter may be defined by:</p> <p style="text-align: center;"><i>email</i></p>
customer_name	<p><b>conditionally absent</b></p> <p>The legally registered name of the Payor (Card Holder).</p> <p>For AVS transactions, this parameter will is always composed in the following format:</p> <p style="text-align: center;"><i>fname lname</i></p> <p>These component strings may be pre-filled in the AVS-enabled API request, or, will be defined when the Payor completes an AVS-enabled Transaction.</p> <p>For non-AVS transactions, this parameter may be defined by:</p> <p style="text-align: center;"><i>name</i></p>
customer_phone	<p><b>conditionally absent</b></p> <p>The contact phone number for the Payor (Card Holder). Please note that for AVS transactions, this parameter will</p>

	<p>always be in E.164 format.</p> <p>This parameter may be pre-filled in the AVS-enabled API request, or, will be defined when the Payor completes an AVS-enabled Transaction.</p> <p>If present in the original request, this parameter may be defined by:</p> <p style="text-align: center;"><i>phone</i></p>
data	<p>The original <i>data</i> submitted in the origin API request.</p> <p>NOTE: This parameter <u>would have been subject to <b>change</b></u>. WiPay's security modules may have either altered or omitted this parameter in the response.</p>
date	<p>An <a href="#">RFC 3339</a>-compliant date and time.</p> <p>This parameter will always have the following format:</p> <p style="text-align: center;"><i>YYYY-MM-DD hh:mm:ss</i></p> <p>Where:</p> <ul style="list-style-type: none"> <li><i>YYYY</i> 4-digit Year</li> <li><i>MM</i> 2-digit Month</li> <li><i>DD</i> 2-digit Day</li> <li><i>hh</i> 2-digit Hour (24-hour)</li> <li><i>mm</i> 2-digit Minute</li> <li><i>ss</i> 2-digit Second</li> </ul>
hash	<p><b>conditionally absent</b></p> <p>A verification check for the response of the Transaction.</p> <p>This <i>hash</i> is calculated using the <a href="#">md5 algorithm</a> on a concatenated string consisting of (in order):</p> <ol style="list-style-type: none"> <li>1. the <i>transaction_id</i></li> <li>2. the original <i>total</i></li> <li>3. the WiPay account's API Key</li> </ol> <p>There are no separators between the strings being concatenated.</p> <p>NOTE: this parameter is returned for <i>status</i> as <i>success</i></p>

	Transactions <b>only</b> .
message	<p>The summary of the Transaction. This parameter is used as the main message and can often offer high-level insight as to what had transpired for any given Transaction.</p> <p>The general format of this parameter will always be:  <math>[ &lt;A&gt;-R&lt;B&gt; ] : &lt;C&gt; .</math></p> <p><b>FAC:</b></p> <p><b>&lt;A&gt;</b> <u>Response Code</u>:</p> <ul style="list-style-type: none"> <li>• "1" - Approved</li> <li>• "2" - Declined</li> <li>• "3" - Error</li> </ul> <p><b>&lt;B&gt;</b> <u>Reason Code</u>: any number between 1 and 9999 (inclusive)</p> <p><b>&lt;C&gt;</b> <u>Reason Description</u>: a short description of the Transaction mainly using the Reason Code Description</p> <p><b>FGB:</b></p> <p><b>&lt;A&gt;</b> <u>Approval Code</u>:</p> <ul style="list-style-type: none"> <li>• "Y" - Approved</li> <li>• "N" - Failed/Error</li> </ul> <p><b>&lt;B&gt;</b> <u>Processor Response Code</u>: any number between 0 and 9999 (inclusive). "00" to "09" (inclusive) is also possible.</p> <p><b>&lt;C&gt;</b> <u>Reason Description</u>: a short description of the Transaction that combines the Approval Code message and the Fail Reason (if it exists).</p>
order_id	<p>The <b>order_id</b> submitted in the original API request.</p> <p>There is no change to this parameter between the request and the response.</p>
status	<p>The result of the Transaction, describing whether it was a <b>success</b>, <b>failed</b> or <b>error</b>.</p> <p>For every completed Transaction, this parameter will always be one of:</p> <ul style="list-style-type: none"> <li>• <b>success</b></li> </ul>

	<ul style="list-style-type: none"> <li>● <b>failed</b></li> <li>● <b>error</b></li> </ul> <p>It is intended that this parameter is to be used together with the <b>message</b> response parameter as the main messaging for the Payor response.</p>
<b>total</b>	<p>The Transaction's final <b>total</b>, i.e. the amount the Payor was debited for.</p> <p>Recall: the final <b>total</b> is primarily affected by <b>fee_structure</b>, where;</p> <ul style="list-style-type: none"> <li>● Under <b>customer_pay</b>, the full Transaction Fee is added to the original total.</li> <li>● Under <b>split</b>, half of the Transaction Fee is added to the original total.</li> <li>● Under <b>merchant_absorb</b>, the original total and final total are the same (no added Transaction Fee).</li> </ul> <p>This final <b>total</b> is also affected by <b>country_code</b>, as described in the <a href="#">Transaction Fee Rates</a> section.</p>
<b>transaction_id</b>	<p>The WiPay Transaction ID for the Transaction.</p> <p>This parameter is very important because WiPay uses this Transaction ID for most of its internal operations after the Payee-Payor Transactional process.</p> <p>This parameter will always consist of the following strings concatenated by a dash ("-") character;</p> <ol style="list-style-type: none"> <li>1. <b>SB</b> prefix (for <b>environment</b> as <b>sandbox</b> Transactions <i>only</i>, not present otherwise)</li> <li>2. A random number between <b>1</b> and <b>99</b></li> <li>3. The Payee's WiPay-internal <b>User ID</b> number</li> <li>4. The <b>order_id</b> submitted in the original API request</li> <li>5. A 24-hour datetime string in the following format: <b>YYYYMMDDhhmmss</b></li> </ol>

## Example

In these examples, we provide the parsed response parameters your web-based application can expect to receive in the querystring of your API request's configured **response\_url**.

## FAC

Non-AVS response parameters would exclude the AVS-only fields. No other differences exist.

In this example, the WiPay Payments Request API request was configured for the TT platform, for a total of 10.00 USD where the customer_pays the Transaction Fees, and AVS set to enabled. On the AVS FAC Hosted Page, all the provided information by the Payor was true and valid, resulting in a successful Transaction.	
card	XXXXXXXXXXXX1111
currency	USD
customer_address	#66 Crossbay Court, Westmoorings, Port Of Spain, Trinidad and Tobago
customer_email	john.doe@example.com
customer_name	John Doe
customer_phone	12462223333
data	"test"
date	2021-06-16 02:41:52
hash	3d34d20260f7433ceee277e9ed9166a3
message	[1-R1]: Transaction is approved.
order_id	oid_123-aBc
status	success
total	12.05
transaction_id	SB-12-1-oid_123-aBc-20210616024001

## FGB

The AVS response parameters do not currently exist for this Hosted Page Service.

In this example, the WiPay Payments Request API request was configured for the JM platform, for a total of 10.00 JMD where the merchant_absorbs the Transaction Fees. On the FGB Hosted Page, an	
--	--

invalid CVV was provided by the Payor for a VISA Credit/Debit Card ending in 0026, resulting in a failed Transaction.	
card	(VISA) ... 0026
currency	JMD
customer_name	John Doe
data	"test"
date	2021-06-16 15:58:54
message	[N-R5101]: 3D Secure authentication failed. Wrong password entered, authentication failed.
order_id	oid_123-aBc
status	failed
total	10.00
transaction_id	SB-84-1-oid_123-aBc-20210616032823

# Transaction Fee Rates

BB	BASIC FREE	Credit/Debit Card	3.80% + Tax <sup>3,4</sup>
GY	NA	Credit/Debit Card	4.20% + GCT <sup>3,4</sup>
JM	BASIC FREE	Credit/Debit Card	4.20% + GCT <sup>3,4</sup>
	Any paid Plan <sup>1</sup>	Credit/Debit Card	3.80% + GCT <sup>3,4</sup>
TT	BASIC FREE	Credit/Debit Card	3.50% + \$0.25 USD <sup>3</sup>
	Any paid Plan <sup>1</sup>	Credit/Debit Card	3.00% + \$0.25 USD <sup>3</sup>

## Notes:

<sup>1</sup> Paid WiPay Plans are monthly (30-day) subscriptions on the WiPay platform that offer certain platform-wide benefits to subscribers.

Paid Plans are (but not limited to);

**BUSINESS:** Business Plus, Business Premium

**PERSONAL:** Personal Premium

<sup>2</sup> Vouchers available only in supported platforms; TT.

<sup>3</sup> Dollar-value fees are relative to the currency of the Transaction and are always converted to the currency-equivalent value for that Transaction.

<sup>4</sup> GCT or Tax refers to “General Consumption Tax” and is applied platform-wide for the BB and JM platforms. It’s dollar-value is calculated as a standard 15% of the total percentage Transactional fee.

# FAQs

## What is the difference between the new and old APIs?

Here, we will compare the Old API and the New API side-by-side. This section will highlight the shortcomings of the Old API and hence show the need for a New API.

	Old API	New API
<b>Accountable by request Origin</b>	No	Yes
<b>AVS capable</b>	No	Yes
<b>Fee Structure supported</b>	No	Yes
<b>JSON response capable</b>	No	Yes
<b>No. countries supported</b>	1	All; BB, GY, JM and TT
<b>No. currencies supported</b>	1	All; JMD, TTD and USD
<b>No. of environments supported</b>	1	All; live and sandbox
<b>Payment method support</b>	1	All; Credit Card, Voucher <sup>1</sup>
<b>Transaction ID assignment</b>	At response	At request
<b>Transaction recovery turnover</b>	Up to 1 hour	5 minutes or less
<b>Other</b>		<ol style="list-style-type: none"><li>1. Enhanced parameter validation (including improved anti-XSS measures)</li><li>2. Enhanced error handling and reporting.</li><li>3. Enhanced fraud protection measures.</li><li>4. Enhanced WiPay Account validation.</li></ol>

<sup>1</sup> Voucher support **coming soon** for supported platforms.

## Request Parameters

Old API	New API (best equivalent)
-	addr1
-	addr2
-	avs
-	city
-	country
-	country_code
-	currency
data	data
developer_id	account_number
email	email
-	environment
-	fee_structure
-	method
name	fname, lname
order_id	order_id
-	origin
phone	phone
return_url	response_url
-	state
total	total
-	zipcode

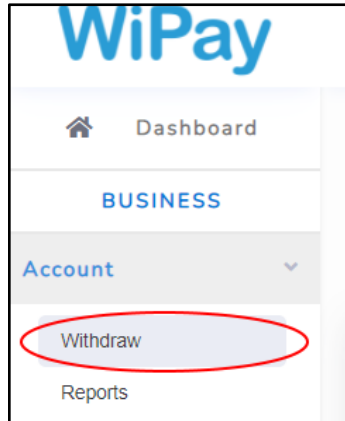
## Response Parameters

Old API	New API (best equivalent)
---------	---------------------------

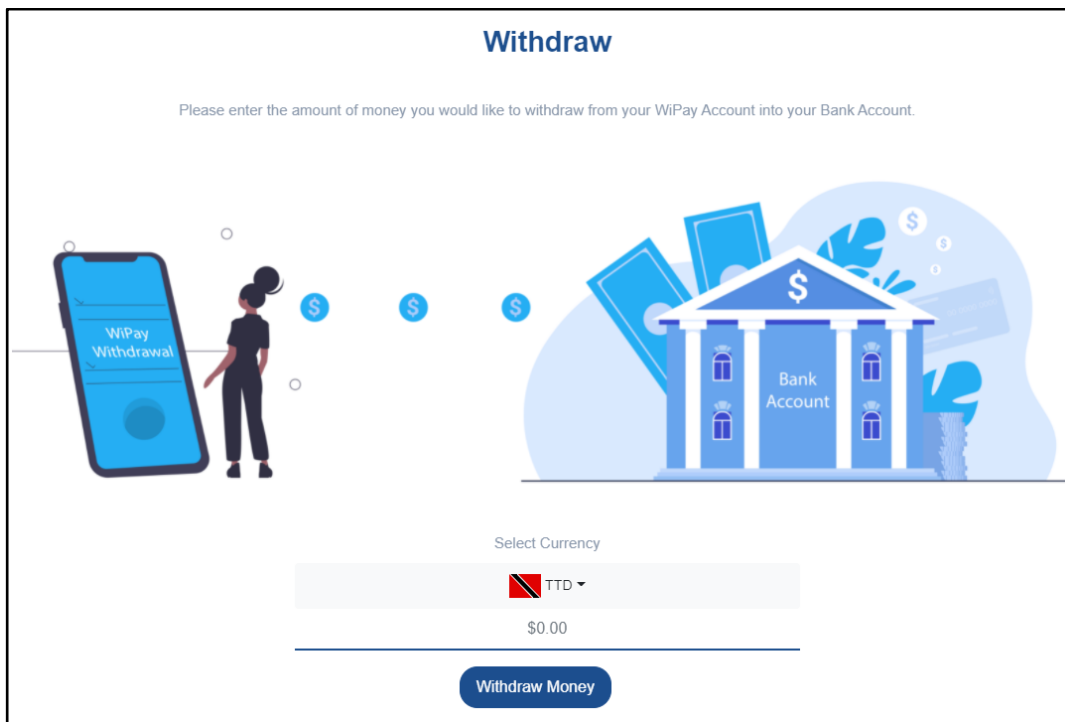
-	card
-	currency
-	customer_address
-	customer_company
D	-
data	data
date	date
email	customer_email
hash	hash
name	customer_name
order_id	order_id
phone	customer_phone
reasonCode	message
reasonDescription	message
responseCode	message
status	status
total	total
transaction_id	transaction_id

## How to get the money into my Bank Account?

Once you are logged into your WiPay Business Account, navigate to your WiPay Account Dashboard. Go to “Account” > “Withdraw” on the left navigation bar.



When you click on “Withdraw”, you will then see this interface:



Follow the instructions on-screen and your Withdrawal request will be made!

Please note:

- All Withdrawals are subject to a flat Withdrawal fee defined by currency and country.
- Withdrawals may take up to **SEVEN (7) working days** to settle into your Bank Account.

- Your WiPay Account must be Verified.
- Your banking information must be accurately and completely entered on your WiPay Account.

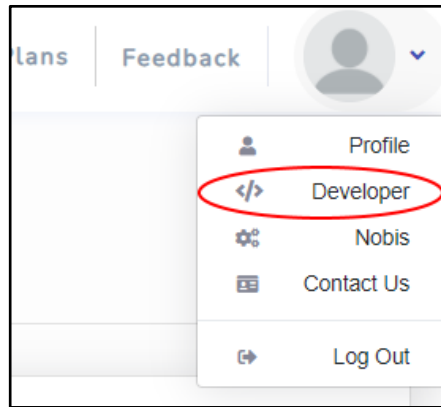
## What is an API Key?

An API Key is a unique alphanumeric string generated by WiPay for your Business Account. The Key acts as a Private Key or “secret component” for the Transaction hashing algorithm. You must use your unique API Key to re-calculate the MD5 hash signature of the Transactions. Transactions are considered authenticated only when the re-calculated MD5 hash signature exactly matches that of the Transaction.

Please do **not** share your API Key. If your API Key has ever been publicly exposed or otherwise compromised, you should [re-generate your API Key](#) as soon as possible.

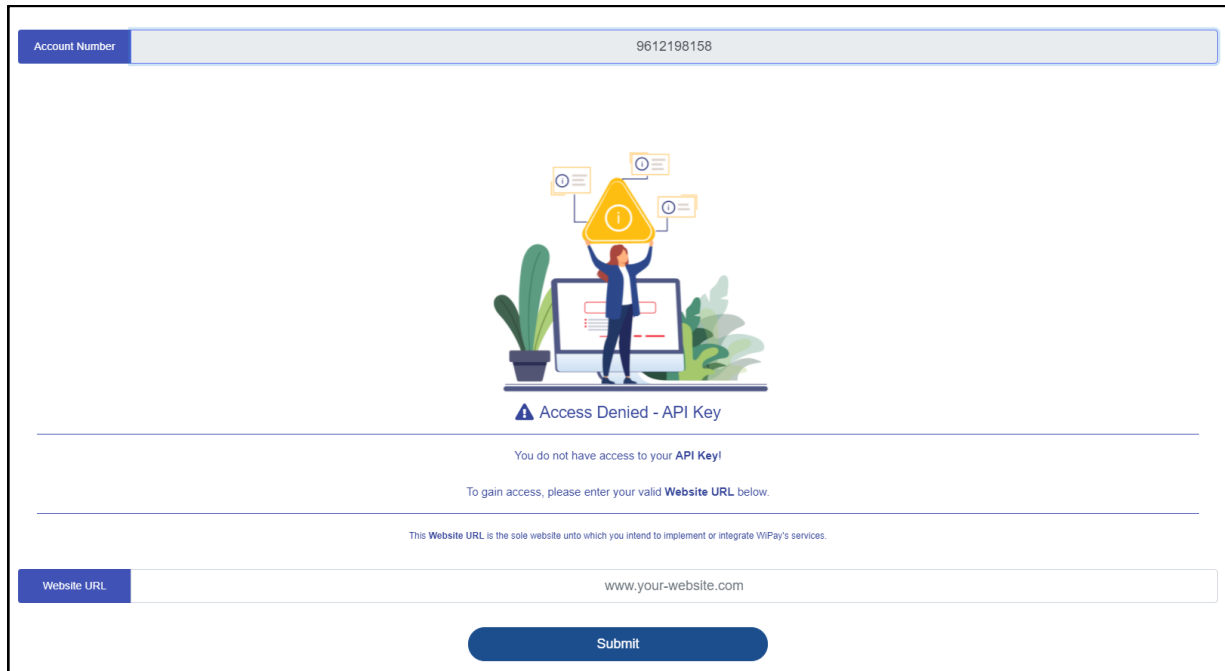
## How to get my API Key?

Once you are logged into your WiPay Business Account, navigate to your WiPay Account Dashboard. Go to the “Developer” section of your Profile by clicking on your Account’s profile picture at the top-right corner of the screen.



NOTE: If you do not see the “Developer” option in the dropdown menu, please ensure that your WiPay Business Account has been Verified.

When you click on “Developer”, you will then see this interface:



Please follow the instructions on screen, and enter the Website URL from which you will be sending your API requests to WiPay. Click “Submit” once done.

Your page will then refresh and you will now see this interface:

Account Number	9612198158
Registered Website	www.example.com
API Key	6bw2j079aau

Press the **Generate new API Key** button below to access a new API Key.

⚠ Generating a **new API Key** will de-register your **old API Key**, and you will lose access to it.

**Generate new API Key**

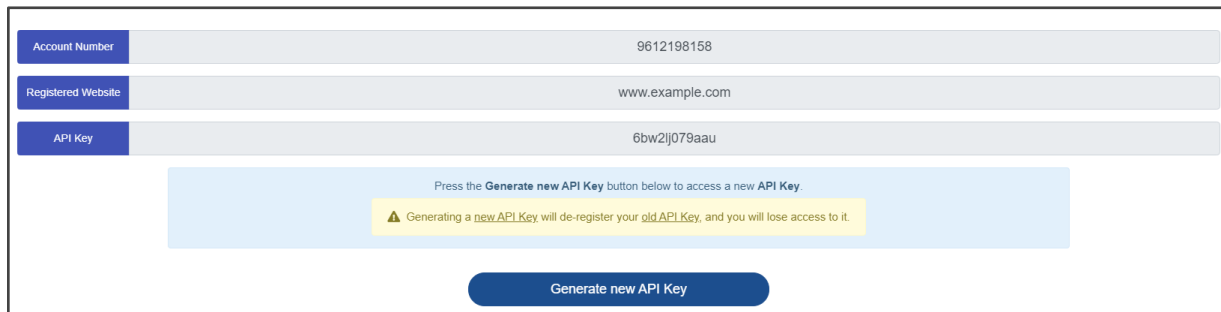
Here, we see that the API Key for this example Account is “6bw21j079aau”.

## How do I generate a new API Key?

Generating a new API Key will permanently unlink your old (current) API Key from your WiPay Business Account. This is especially useful if your existing API Key was compromised in any way.

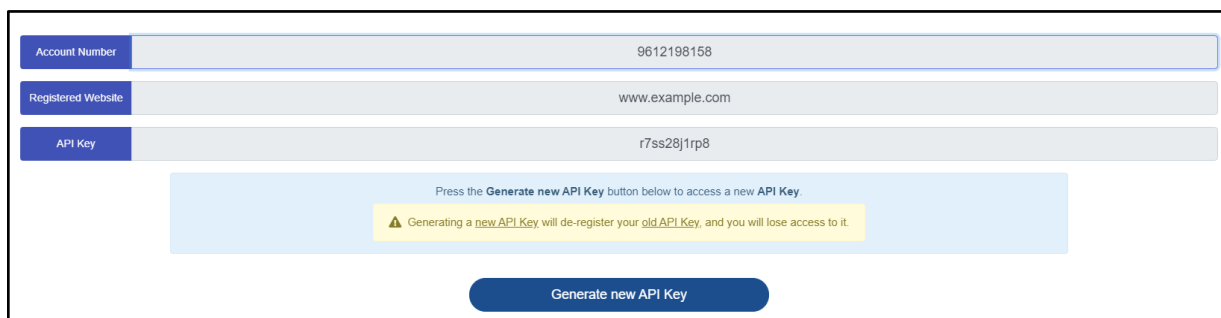
Once you are logged into your WiPay Business Account, navigate to your WiPay Account Dashboard. Go to the “Developer” section of your Profile by clicking on your Account’s profile picture at the top-right corner of the screen.

If you already [have an API Key](#), you will see this interface:



The screenshot shows a user interface with three rows of account information: Account Number (9612198158), Registered Website (www.example.com), and API Key (6bw21j079aau). Below this is a light blue box containing a warning message: "Generating a new API Key will de-register your old API Key, and you will lose access to it." At the bottom of this box is a dark blue button labeled "Generate new API Key".

Click on the “Generate new API Key” button, and hit “Generate New API Key” on the pop-up dialog. WiPay will automatically generate a new API Key for you account, unlink the old API Key and use this new API Key for all operations henceforth. Your page should automatically reload and reflect your new API Key.



This screenshot is identical to the previous one, but the API Key field now displays "r7ss28j1rp8", indicating that a new key has been generated and the old one has been replaced.

Here we see that the API Key for this example Account was changed from “6bw21j079aau” to “r7ss28j1rp8”.

## What is the API Key of the TEST Account?


The API Key of the TEST Account is 123.

# Appendices

## Appendix 1 - Other FGB pages

### Success transaction summary

Version: 21.3.0-302      Build time: 2021-06-09T17:21:46+0200      Last commit: 420a694



**Amount \$ 10.00 JMD** First Global hosted payment page

Order Confirmation

Transaction State	APPROVED
Order id	SB-91-1-test-20210616033704
Time	16/06/21 22:37:54
Ref. No.	84565235227
Approval Code	Y:OK6879:4565235227:PPXX:089311
Total	10.00
Currency	JMD

Please retain this copy for statement verification.

[Return to Shop](#)

---

We recommend that you print this page as confirmation of your order

## Fail transaction summary

Version: 21.3.0-302

Build time: 2021-06-09T17:21:46+0200

Last commit: 420a694



Your Commercial Bank from GraceKennedy

**Amount \$ 10.00 JMD**

First Global hosted payment page

### Order Confirmation

Transaction State	FAILED
Error	Wrong password entered, authentication failed.
Order id	SB-84-1-test-20210616032823
Time	16/06/21 22:29:38
Approval Code	N:-5101:3D Secure authentication failed
Total	10.00
Currency	JMD


Please retain this copy for statement verification.

[Return to Shop](#)

---

We recommend that you print this page as confirmation of your order

## 3DS Authentication password dialog




### Authenticate using password

Please enter your password below.

Merchant: WIPAY  
Amount: JMD 10.00  
Date: 20210616 20:39:44  
Card Number: XXXX XXXX XXXX 4977  
Personal Message: Secret!33  
Password:

[Help](#) [Cancel](#)

If you experience any problems, please contact our Helpdesk on the back of your card.



### Authenticate using password

Please enter your password below.

Merchant: WIPAY  
Amount: JMD 10.00  
Date: 20210617 03:56:23  
Card Number: XXXX XXXX XXXX 1003  
Personal Message: Secret123  
Password:

[Help](#) [Cancel](#)

If you experience any problems, please contact our Helpdesk on the back of your card.