

mcharge mobile-web payments

Interface Specifications

Version 1.1.2

31/05/2011



1. Revisions

Rev#	Date	Remark
1.0	07-04-2008	Initial version
1.1	21-04-2008	Adding advanced parameters to the startTransaction method Adding support for a subscription-based mobile-web payment model
1.1.1	19-05-2008	Adding parameters to startTransaction and before Subscriptions can be used
1.1.2	22-05-2008	Shortening the text used for subscription's reminder SMS

2. Conventions

This document describes the interface specifications for the Mpulse Gateway SOAP Web Service interface. The Mpulse Gateway provides a Web Service Description Language (WSDL) document that describes the web service. The web service binding is configured as follows:

- Service name: WapBilling
- Namespace: http://billing.mpulse.eu/wapbilling
- Style: RPC
- Use: Literal

We use this SOAP binding mode because it's known to be compatible with most available SOAP/WSDL clients on the market – as opposed to Document/Literal/Wrapped combinations mostly used by Microsoft-based applications.

3. Authentication

The web service endpoint uses HTTP basic authentication to identify the user account calling its methods. Access to the WSDL document is not protected.

For SOAP clients that do not yet support HTTP basic authentication for web service endpoint, a different version of the endpoint is available that uses explicit authentication through method parameters. Please contact your support team for further information.

4. Single Transactions

Single (or one-off) transactions are used for simple purchase orders and are not recurrent. Hence, the interaction between the content provider and the Mpulse Gateway is rather simple too and is limited to two synchronous method calls to initiate the transaction (startTransaction) and to retrieve the status of the transaction (getTransactionStatus) when the end-user has either confirmed or cancelled the request. Please refer to section 4.4 for more details.

Before Single Transactions can be used, some default parameters have to be configured by the Mpulse Sales team. The following information is required:

- **providerId** – a unique identification token for the content provider (can either be the provider's name or the portal address)
- **support** – end-user support contact details (can either be an email address or a telephone number)
- **redirectUrl** – the URL where the end-user should be redirected to after confirming the order
- **template** – the name of the layout template that should be used to display the confirmation page. New customized templates can be added on demand.



- **language** – the default language that should be used to display the confirmation page

4.1. startTransaction

The startTransaction method is called by the content provider when the end-user has selected a product that he wants to purchase, but before he's forwarded to the Mpulse Gateway page to confirm his choice. All required information is passed to the Mpulse Gateway and a unique transactionId is returned that can be used at a later time to retrieve the status of the transaction.

The method has the following parameters:

- **product** (xsd:string) – the name of the product or service that is about to be purchased
- **price** (xsd:int) – the end-user price including taxes expressed in Euro-cents
- **age** (xsd:int) – the age limit for this product or service if applicable; accepted values: 0 (no limit), 12, 14, 16, 18

It will return the following values wrapped in a Response instance:

- **transactionId** – a unique transaction ID that is assigned by the gateway and can be reused later to retrieve the status of the transaction
- **redirectUrl** – the URL of the confirmation page that the end-user's browser should be redirected to in order to complete the transaction

In case of an error, the following fault types can be returned:

- **AuthenticationException** – occurs if no valid mobile-web payment account can be found with the credentials provided through the HTTP Basic Authentication mechanism or the account has not sufficient permissions to execute this method
- **BillingException** – occurs if the validation of the method parameters fails

In both cases a message giving more details about the error is provided with the fault message.

4.2. startTransactionAdvanced

Certain default parameters (initially configured) may be overridden at the transaction level. The startTransactionAdvanced method does exactly the same thing than the startTransaction method, except that it allows overriding of the following parameters:

- **providerId** (xsd:string) – a unique identification token for the content provider (can either be the provider's commercial name or the portal address)
- **redirectUrl** (xsd:string) – the URL of the content provider that the end-user needs to be redirected to after the transaction has been completed
- **template** (xsd:string) – the name of the template that should be used to display the confirmation dialog to the end-user
- **language** – The language of the confirmation page (possible values: ENGLISH, FRENCH, GERMAN)

The return and fault types of startTransactionAdvanced are identical to startTransaction.

4.3. getTransactionStatus

The `getTransactionStatus` method is called after end-user confirmation of the transaction. The content provider would call this method to find out whether the payment was successful or not and whether the purchased product or service can be delivered.

The method takes a single parameter:

- **transactionId** (xsd:string) – the transaction ID received in the response to the `startTransaction` call

It returns a `TransactionStatus` instance with the following information:

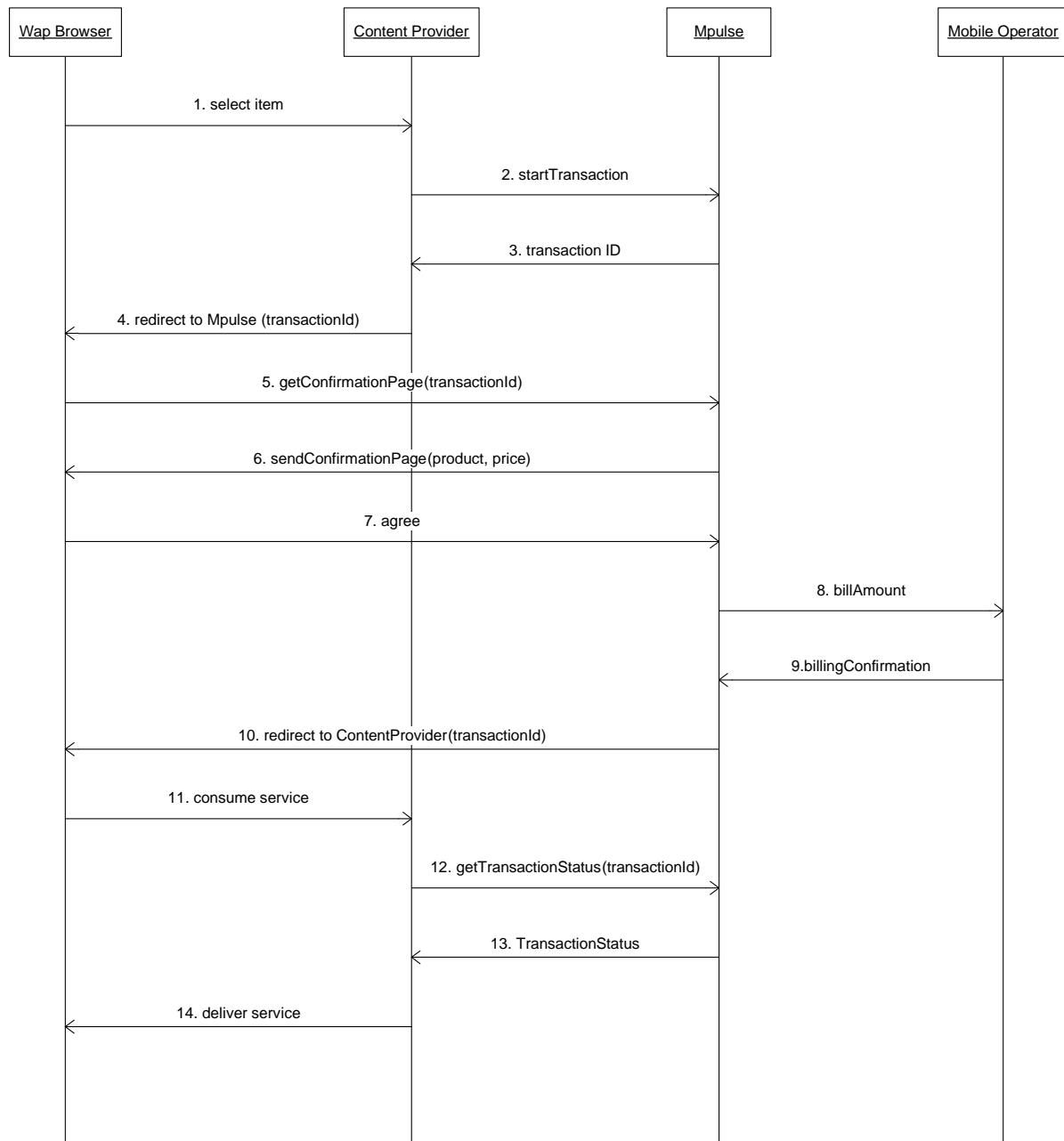
- **status** (xsd:status) – an enumeration type that can take the following values:
 - **PENDING** – the end-user has neither confirmed nor cancelled the transaction on the confirmation page
 - **SUCCESS** – the end-user has confirmed the transaction and the payment has been successfully executed
 - **ERROR** – the end-user has confirmed the transaction, but there was an error during the payment procedure
 - **CANCEL** – the end-user has chosen not to confirm the transaction but pressed the Cancel button instead
- **statusMessage** (xsd:string) – contains more information in case of an error
- **timestamp** (xsd:dateTime) – the date and time of the transaction
- **amount** (xsd:int) – the amount that was actually billed to the end-user; in the great majority of the cases, this amount corresponds to the amount specified during the `startTransaction` method call, but in some rare cases, the amount has to be split up into several billing operations on the mobile operator side; if some but not all of them succeed, status will be **SUCCESS**, but amount will be less than initially requested
- **msisdn** (xsd:string) – the MSISDN is returned for further reference and can be used as an optional argument to `startTransaction` in order to avoid complex lookup mechanisms using the IP address

In case of an error, the following fault types can be returned:

- **AuthenticationException** – occurs if no valid mobile-web payment account can be found with the credentials provided through the HTTP Basic Authentication mechanism or the account has not sufficient permissions to execute this method
- **BillingException** – occurs if the transaction ID is invalid or it does not belong to a transaction initiated by the same account

4.4. Message flow

The following diagram describes the message and information flow of a successful mobile-web payment transaction and consists of 14 steps.



1. The end-user selects the item that he wants to purchase (a) on the content provider's WAP page, or (b) by sending an SMS to the content provider.
2. The content provider initiates a mobile-web payment transaction by calling the startTransaction method and passing all required parameters such as product name, price, age limit etc.
3. The Mpulse Gateway will generate a unique transaction ID and return it to the content provider, together with the URL of the confirmation page that the content provider will redirect the end-user to.

4. The Content provider (a) redirects the end-user that is browsing its WAP portal to the Mpulse Gateway confirmation page, or (b) sends a WAP-Push to the end-user, pointing towards the Mpulse Gateway confirmation page.
5. The end-user requests the confirmation page using the transaction ID. Mpulse Gateway displays the confirmation page with the providerId, product, price, age and support parameters from the startTransaction method, and with the options to confirm or to cancel the transaction.
6. The end-user hits the Confirm button.
7. The Mpulse Gateway identifies the operator of the end-user using his IP address and/or additional information in the request headers and looks up the MSISDN. It then executes a billing request on the corresponding operator's billing gateway.
8. The Mpulse Gateway saves the response message of the external billing gateway with the current transaction information.
9. The end-user is redirected to the content provider's content delivery page.
10. The end-user tries to access the content that he has purchased.
11. The content provider retrieves the necessary status information about the transaction using the getTransactionStatus method.
12. The Mpulse Gateway returns the status of the transaction.
13. The content provider decides whether to deliver or not the requested content, depending on the information contained in the transaction status message.

5. Subscriptions

As opposed to single payments, end-users can also sign up for a service subscription which will automatically be renewed by the Mpulse Gateway. Before subscriptions can be used, they have to be configured by the Mpulse Sales team. The following information is required:

- **providerId** – a unique identification token for the content provider (can either be the provider's name or the portal address)
- **name** – a short identifier for this subscription (e.g. GAMES, MUSIC, SEXY...)
- **shortcode** – the shortcode used by end-users to unsubscribe. By default this is a shared shortcode (64111 in Luxembourg). When using a dedicated shortcode for a country where unsubscribing must be free of charge to the user (namely in Luxembourg), then: (a) if a Premium MT shortcode is selected, confirmation SMS MT are sent to users via a shared (non-Premium) shortcode (64111 in Luxembourg), and (b) Premium MO shortcodes may not be selected.
- **price** – the end-user price including taxes expressed in Euro-cents (maximum 1200 cents)
- **age** – the age limit for this product or service if applicable; accepted values: 0 (no limit), 12, 14, 16, 18
- **description** – a description of the service that the user is subscribing to
- **interval** – the interval of the subscription renewals expressed in days or months
- **support** – end-user support contact details; can either be an email address or a telephone number

Note that the content provider's marketing material must state the *providerId*, *name* and the unsubscription method (user sends the text "STOP *name*" via SMS MO to the *shortcode*). The confirmation page will display all of the parameters above (*interval* may be rounded to weeks).

Users receive a reminder of the subscription once per month, via SMS MT (free to the user). For each user, the content provider chooses the language of the reminder SMS when calling the `startSubscription` method. The text used for reminder SMS is the following:

- **German** - Sie sind Mitglied des *name* Clubs von *providerId* für *priceEUR/interval*. Zum Stoppen STOP *name* an *shortcode* schicken. Helpdesk: *support*
- **French** - Tu es inscrit au Club *name* de *providerId* pour *priceEUR/interval*. Pour quitter envoi STOP *name* par SMS au *shortcode*. Helpdesk: *support*

Bear in mind when choosing these parameters that if the resulting text is over 160 characters, the content provider's cost of sending reminder SMS will double.

5.1. startSubscription

- **name** (xsd:string) – a short identifier for this subscription (e.g. GAMES, MUSIC, SEXY...)
- **startDate** (xsd:dateTime) – the date and time of the first transaction associated with this subscription – if left blank, the first transaction will be executed when the user confirms his subscription
- **endDate** (xsd:dateTime) – the date and time of the last transaction associated with this subscription – if left blank, there will be no automatic end-date of the subscription. The end-user has to manually sign out

It will return the following values wrapped in a Response instance:

- **transactionId** – a unique transaction ID that is assigned by the gateway and can be reused later to retrieve the status of the transaction
- **redirectUrl** – the URL of the confirmation page that the end-user's browser should be redirected to in order to complete the transaction

In case of an error, the following fault types can be returned:

- **AuthenticationException** – occurs if no valid mobile-web payment account can be found with the credentials provided through the HTTP Basic Authentication mechanism or the account has not sufficient permissions to execute this method
- **BillingException** – occurs if the validation of the method parameters fails

In both cases a message giving more details about the error is provided with the fault message.

5.2. startSubscriptionAdvanced

Certain default parameters (initially configured) may be overridden on a subscription level. The `startSubscriptionAdvanced` method does exactly the same thing than the `startSubscription` method, except that it will allow overriding of the following parameters:

- **redirectUrl** (xsd:string) – the URL of the content provider that the end-user needs to be redirected to after the transaction has been completed
- **template** (xsd:string) – the name of the template that should be used to display the confirmation dialog to the end-user
- **language** – The language of the confirmation page (possible values: ENGLISH, FRENCH, GERMAN)

The return and fault types of `startSubscriptionAdvanced` are identical to `startSubscription`.

5.3. **getSubscriptionStatus**

The `getSubscriptionStatus` method can be called at any moment by the content provider to retrieve the current status of the subscription.

The method takes a single parameter:

- **transactionId** (xsd:string) – the transaction ID received in the response to the `startSubscription` call

It returns a `SubscriptionStatus` instance with the following information:

- **status** (xsd:status) – an enumeration type that can take the following values:
 - ACTIVE – the subscription is active and renewals have been successful
 - ACTIVE_RETRY – the subscription is active but the last transaction has failed and is scheduled for retry
 - CANCEL – the transaction has been refused by the end-user on the confirmation screen
 - STOPPED – the end-user has explicitly requested this subscription to be stopped
- **statusMessage** (xsd:string) – contains more information in case of an error
- **msisdn** (xsd:string) – if the content provider's account is set up accordingly, the MSISDN if the end-user is returned for further reference and can be used as an optional argument to `startSubscription` in order to avoid complex lookup mechanisms using the IP address
- **startDate** (xsd:dateTime) – the date and time of the first transaction associated with this subscription
- **endDate** (xsd:dateTime) – the date and time of the last transaction associated with this subscription.
- **lastTransaction** – the transaction status of the last transaction that was executed as returned by `getTransactionStatus()`.

In case of an error, the following fault types can be returned:

- **AuthenticationException** – occurs if no valid mobile-web payment account can be found with the credentials provided through the HTTP Basic Authentication mechanism or the account has not sufficient permissions to execute this method
- **BillingException** – occurs if the transaction ID is invalid or it does not belong to a transaction initiated by the same account

5.4. **getSubscriptionTransactions**

The `getSubscriptionTransactions` method can be called if detailed information about the individual transactions that have been performed in the context of this subscription.

The method takes a single parameter:

- **transactionId** (xsd:string) – the transaction ID received in the response to the `startSubscription` call

The method returns a list of `TransactionStatus` instances as defined in section 4.3. All the transactions from the beginning of the subscription until the current date or the end of the subscription (if applicable) will be returned.

In case of an error, the following fault types can be returned:

- **AuthenticationException** – occurs if no valid mobile-web payment account can be found with the credentials provided through the HTTP Basic Authentication mechanism or the account has not sufficient permissions to execute this method
- **BillingException** – occurs if the transaction ID is invalid or it does not belong to a transaction initiated by the same account

5.5. **getSubscriptionTransactionsBetween**

The `getSubscriptionTransactionsBetween` does the same thing as `getSubscriptionTransactions` except that it takes 2 additional parameters to specify the start and end dates of the transactions to be returned:

- **transactionId** (xsd:string) – the transaction ID received in the response to the `startSubscription` call
- **startDate** (xsd:dateTime) – the date and time of the last transaction to be returned for this subscription
- **endDate** (xsd:dateTime) – the date and time of the last transaction to be returned for this subscription

The method returns a list of `TransactionStatus` instances as defined in section 4.3. All the transactions between start date and end date will be returned.

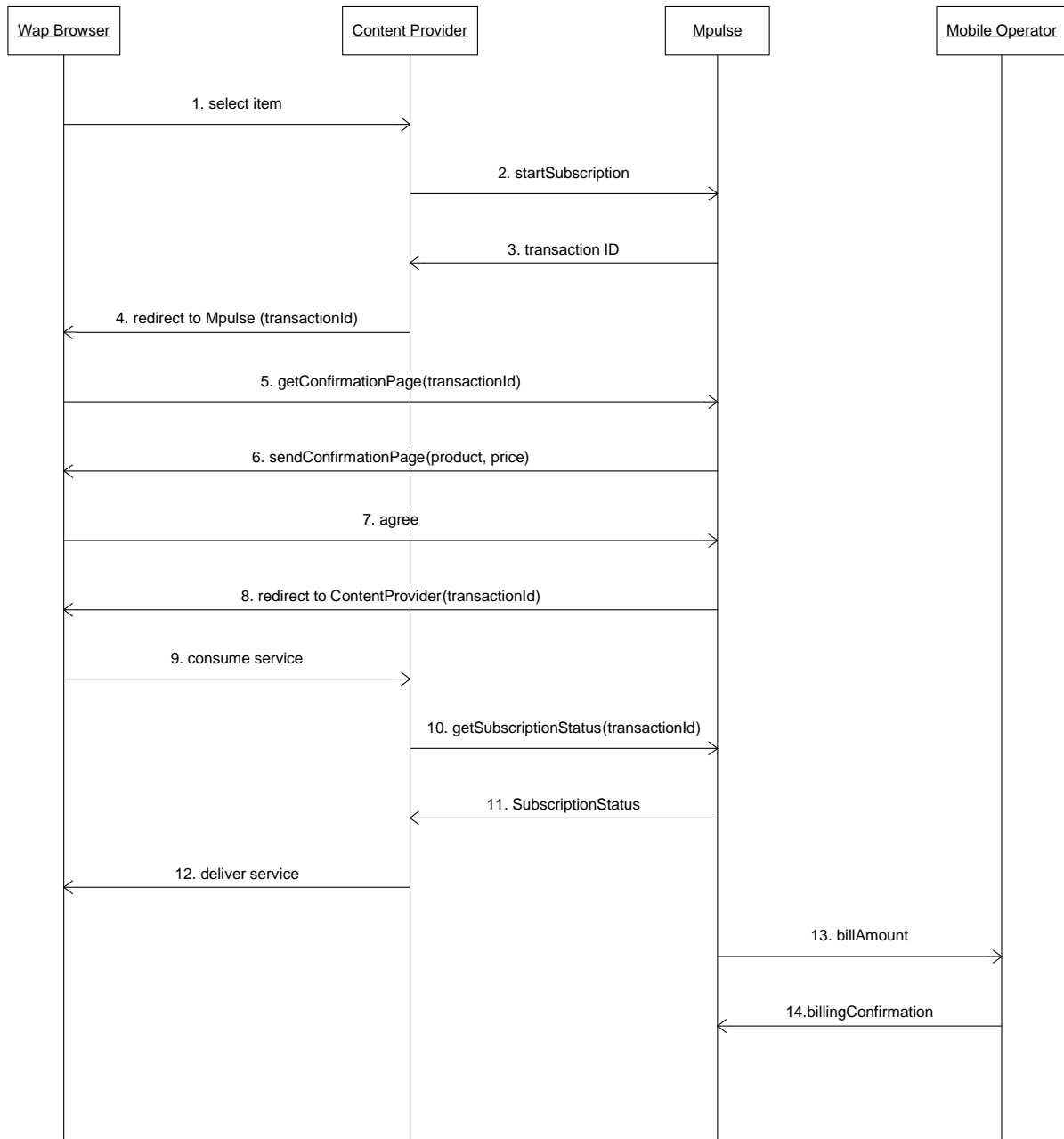
In case of an error, the following fault types can be returned:

- **AuthenticationException** – occurs if no valid mobile-web payment account can be found with the credentials provided through the HTTP Basic Authentication mechanism or the account has not sufficient permissions to execute this method
- **BillingException** – occurs if the transaction ID is invalid or it does not belong to a transaction initiated by the same account

5.6. **Message flow**

The following message flow describes a successful service subscription including billing operations with the external billing providers.

The work-flow is very similar to the single transactions, except that a subscription consists of recurring transactions that belong to a single service. Hence, steps 10&11 and 13&14 can be repeated as many times as necessary and until the end-user decides to cancel the subscription or until the end of the subscription term has been reached.



6. Redirection

In order for the scenarios above to work, the mobile browser needs to be redirected to the Mpulse Gateway Billing confirmation screen and, after the transaction has been completed, back to the content provider's page.



The `startTransaction()` and `startSubscription()` methods return the redirect URL that the content provider must use to redirect to the validation screen of the Mpulse Gateway and it already includes the transaction ID in the URL encoded as HTTP GET parameter.

The same technique will be used when the WAP browser is redirected back to the content provider's WAP portal. The Mpulse Gateway will use the redirect URL that has been assigned to the content provider's account at setup time or use the one that has explicitly been defined in the `startTransactionAdvanced()` or `startSubscriptionAdvanced()` methods and add a HTTP GET parameter called 'tid' that contains the value of the transaction ID belonging to the transaction of the user.

This transaction ID can be used to retrieve the status of the transaction or subscription using `getTransactionStatus()` or `getSubscriptionStatus()` respectively.