

mcharge web payments

# Internet+ Interface Specifications for France

Version 1.0  
31/05/2011



## 1. Revisions

Rev#	Date	Remark
1.0	25/03/2010	Initial review based on mobile web payment workflows

## 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: InternetPlus
- Namespace: <http://billing.mpulse.eu/internetplus>
- 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)
- **logo** – a 100x40 pixels GIF file that's displayed on the payment screen

### 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 confirmation page (hosted on the Mpulse Gateway or on the operator's portal depending on the operator configuration) 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 second parameter the method will return is a redirectUrl. This is the location of the confirmation page that the end-user should be redirected to using an HTTP redirect. The URL could be a URL hosted by Mpulse or a URL hosted by the mobile operator, depending on the workflows used by the operator.

The method has the following parameters:

- **product** (xsd:string) – the name or identifier 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. Today, services for the 18+ age group are not permitted on the Internet+ platform.
- **ipAddress** (xsd:string) – The IP address (in dot-decimal notation) of the mobile client. This parameter is used to determine the operator of the client and compute an appropriate redirectUrl. If ipAddress is empty, the user will have to be redirected to an intermediate Mpulse page that will detect the operator and forward them to the right address for the next step of the flow.
- **redirectUrl** (xsd:string) – the URL of the content provider that the end-user needs to be redirected to after the transaction has been completed

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 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. 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 when the end-user is accessing the content page of the content provider provided in the startTransaction call as redirectUrl parameter.

The method takes a single parameter:

- **transactionId** (xsd:string) – the transaction ID received in the response of 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

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

- **AuthenticationException** – occurs if no valid 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

### 4.3. confirmTransaction

After content delivery, the content provider needs to confirm the transaction to commit the payment on the operator's side. Without confirmation, the transaction will generate no revenue. The confirmation cannot occur more than 24 hours after the beginning of the transaction, otherwise it will automatically be cancelled.

The method takes a single parameter and returns nothing (void):

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

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

- **AuthenticationException** – occurs if no valid 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, or if the confirmation is rejected by the operator.

### 4.4. cancelTransaction

If, for some reason, the ordered content cannot be delivered to the end-consumer, the content provider can either choose to do nothing (no confirmTransaction) and the transaction will get cancelled automatically after 24 hours, or call cancelTransaction explicitly. The end-consumer will not be charged for the service.

The method takes a single parameter and returns nothing (void):

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

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

- **AuthenticationException** – occurs if no valid 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, or if the cancellation is rejected by the operator.

#### 4.5. refundTransaction

If, by mistake or other reason, a transaction has been confirmed and it turns out that the end-consumer cannot use the delivered content or service or claims a refund, this method can be used. After refunding a transaction it cannot be confirmed or cancelled anymore. The amount of the transaction will be debited again from the content provider's account and credited back to the end-consumer.

The method takes a single parameter and returns nothing (void):

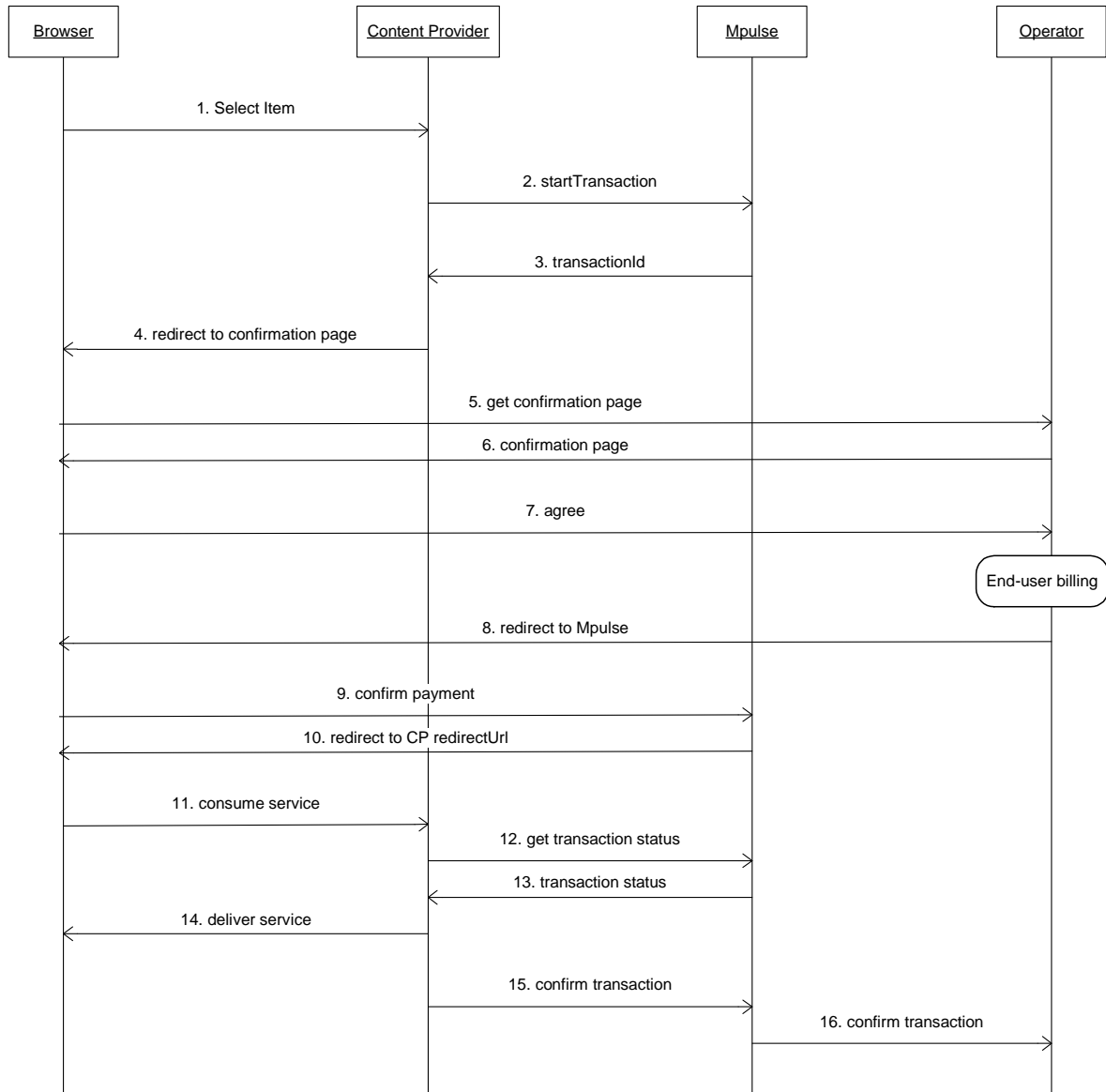
- **transactionId** (xsd:string) – the transaction ID received in the response of the startTransaction call

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

- **AuthenticationException** – occurs if no valid 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, or if the cancellation is rejected by the operator.

#### 4.6. Message flow

The following diagram describes the message and information flow of a successful web payment transaction and consists of 16 steps.



1. The end-user selects the item that he wants to purchase on the content provider's web page
2. The content provider initiates a 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. This page may be hosted by Mpulse or by the operator himself, depending on the workflows in place
4. The Content provider redirects the end-user that is browsing its web portal to the confirmation page using a HTTP redirect
5. The end-user requests the confirmation page using the transaction ID and/or additional parameters.

6. The operator 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.
7. The end-user hits the Confirm button and is the amount of the transaction charged to his account.
8. The operator redirects the end-user to an intermediate page hosted by Mpulse in order to confirm the transaction status.
9. Mpulse will then update the transaction database with the new status.
10. The Mpulse Gateway will then redirect the end-user to the content provider's redirect URL with the content delivery page that was provided in the startTransaction call (step 2)
11. The end-user tries to access the content that he has purchased.
12. The content provider retrieves the necessary status information about the transaction using the getTransactionStatus method.
13. The Mpulse Gateway returns the status of the transaction.
14. The content provider decides whether to deliver or not the requested content, depending on the information contained in the transaction status message.
15. After content delivery, the content provider will have to confirm the transaction in order to receive the payment. This step is mandatory.
16. Mpulse transmits the transaction confirmation to the operator.

## 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)
- **subscriptionName** – a short identifier for this subscription (e.g. GAMES, MUSIC, SEXY...)
- **price** – the end-user price including taxes expressed in Euro-cents
- **age** – the age limit for this product or service if applicable; accepted values: 0 (no limit), 12, 14, 16, 18. Today, services for the 18+ age group are not permitted on the Internet+ platform.
- **description** – a description of the service that the user is subscribing to
- **interval** – the interval of the subscription renewals. Subscriptions can be renewed weekly or monthly
- **support** – end-user support contact details; can either be an email address or a telephone number

### 5.1. startSubscription

- **subscriptionName** (xsd:string) – a short identifier for this subscription (e.g. GAMES, MUSIC, SEXY...)
- **redirectUrl** (xsd:string) – the URL of the content provider that the end-user needs to be redirected to after the subscription sign-up has been completed

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

- **subscriptionId** – a unique subscription 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 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 (e.g. the *name* parameter is invalid or has not been defined before)

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

## 5.2. **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:

- **subscriptionId** (xsd:string) – the subscription ID received in the response of the `startSubscription` call

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

- **status** (xsd:status) – an enumeration type that can take the following values:
  - PENDING – the end-user has taken no action on the confirmation screen yet
  - ACTIVE – the subscription is active and renewals have been successful
  - CANCEL – the subscription has been refused by the end-user on the confirmation screen
  - STOPPED – the end-user has explicitly requested this subscription to be stopped
  - STOP\_REQUESTED – the user has requested his subscription to be stopped, but it has not yet become effective
- **statusMessage** (xsd:string) – contains more information in case of an error
- **endDate** (xsd:dateTime) – the date at which the cancellation of the subscription has been requested or has become effective (if status = STOP\_REQUESTED or STOPPED respectively, empty otherwise)

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

- **AuthenticationException** – occurs if no valid 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 subscription ID is invalid or it does not belong to a transaction initiated by the same account

## 5.3. **confirmSubscription**

After the subscription has been started, the content provider needs to explicitly confirm this by calling this method. Only then charging of the end-user can begin.

The method takes a single parameter and returns nothing (void):

- **subscriptionId** (xsd:string) – the subscription ID received in the response of the startSubscription call

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

- **AuthenticationException** – occurs if no valid 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 subscription ID is invalid or it does not belong to a transaction initiated by the same account

## 5.4. cancelSubscription

If, for some reason, the subscription cannot be started, the content provider can either choose to do nothing (no confirmTransaction) and the subscription will get cancelled automatically after 24 hours, or call cancelTransaction explicitly. The end-consumer will not be charged for the service.

The method takes a single parameter and returns nothing (void):

- **subscriptionId** (xsd:string) – the subscription ID received in the response of the startSubscription call

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

- **AuthenticationException** – occurs if no valid 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 subscription ID is invalid or it does not belong to a transaction initiated by the same account

## 5.5. stopSubscription

This method must be called if the user selects the option to cancel the subscription. It takes only one parameter and returns no value.

- **subscriptionId** (xsd:string) – the subscription ID received in the response of the startSubscription call
- **immediate** (xsd:boolean) – defines whether the subscription needs to be cancelled immediately (true) or if we should wait until the end of the minimum subscription period that has been agreed with the end-user, if it exists, or at the end of the current period if it doesn't exist (false).
- **userComment** (xsd:string) – a comment about why the user requested that his subscription is stopped
- **adminComment** (xsd:string) – a comment about why the content provider wanted to cancel the subscription
- **reasonCode** (xsd:int)
  - **0:** generic reason
  - **1:** multiple subscription to the same service
  - **2:** multiple invoicing for the same service
  - **3:** service not delivered
  - **4:** purchase disputed by the end-user
  - **5:** child purchase without parent consent

- **6:** transaction for a subscription that had already been cancelled

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

- **AuthenticationException** – occurs if no valid 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 subscription ID is invalid or it does not belong to a transaction initiated by the same account

## 5.6. refundSubscription

If, by mistake or other reason, a subscription has been confirmed and billed one or more times and it turns out that the end-consumer cannot use the delivered content or service or claims a refund, this method can be used.

The method takes a single parameter and returns nothing (void):

- **subscriptionId** (xsd:string) – the subscription ID received in the response of the startSubscription call
- **userComment** (xsd:string) – a comment about why the user requested that his subscription is stopped
- **adminComment** (xsd:string) – a comment about why the content provider wanted to cancel the subscription

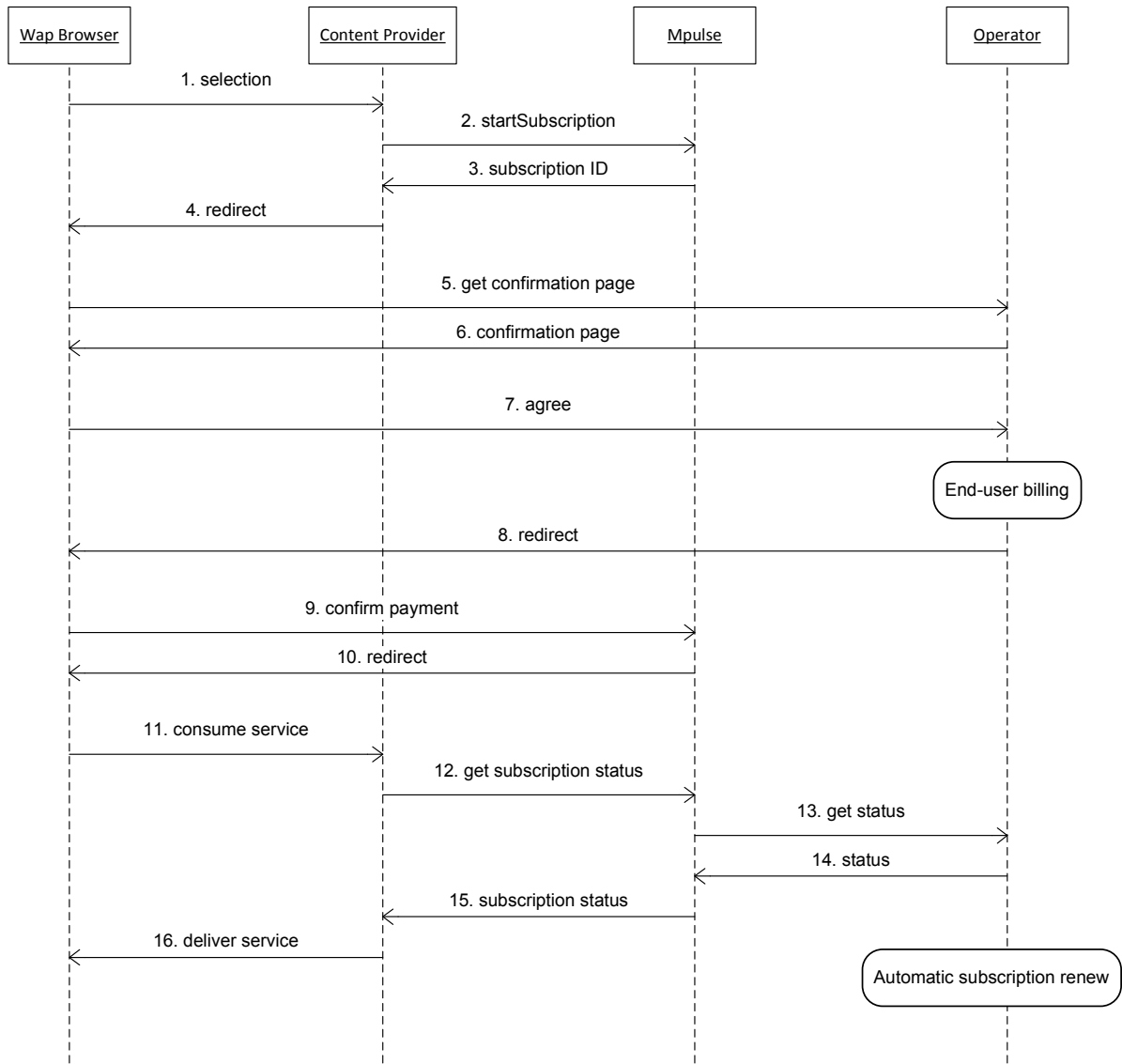
In case of an error, the following fault types can be returned:

- **AuthenticationException** – occurs if no valid 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 subscription ID is invalid or it does not belong to a transaction initiated by the same account

## 5.7. Message flow

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

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 11&12 and 15&16 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 above scenarios to work, the browser needs to be redirected to a 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 operator and it already includes the transaction ID and other parameters depending on the mobile operator in the URL encoded as HTTP GET parameters.

The same technique will be used when the web browser to redirected back to the content provider's web portal. The Mpulse Gateway will use the redirect URL that has defined in the `startTransaction` or `startSubscription` 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.

