

mcharge mobile-web payments

Interface Specifications for France

Version 1.2.1
31/05/2011



1. Revisions

Rev#	Date	Remark
1.0	04/03/2010	Fork the Luxembourg workflow and add functionality to support the workflows of French operators
1.1	08/26/2010	Adding an IP Address parameter to startSubscription (inspired by startTransaction) to save an extra redirection.
1.2	11/03/2011	Implement the new cancelSubscription process on Bouygues
1.2.1	29/04/2011	Adapt cancelSubscription return values to match other methods

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 4.3 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)
- **template** – the name of the layout template that should be used to display the confirmation page. New customized templates can be added on demand.

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
- **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 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. 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

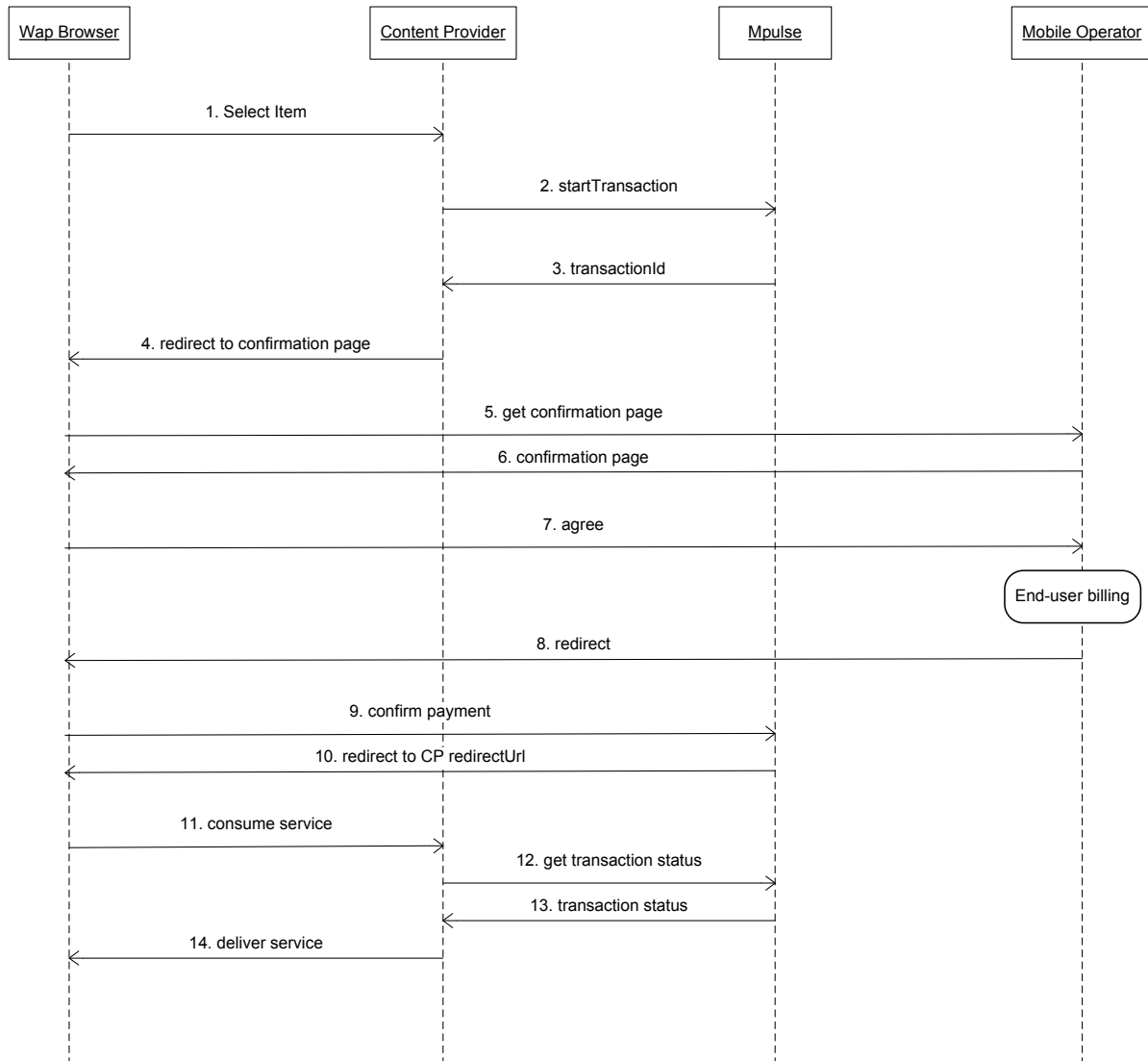
It returns a TransactionStatus instance with the following information:

- **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 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

4.3. 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 on the content provider’s WAP page
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. 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 WAP 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. Mpulse Gateway (or the mobile operator, depending on the scenario) 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.

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...)
- **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
- **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

- **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
- **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.

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 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 (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. 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 subscription sign-up has been completed

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:

- **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:
 - 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
- **statusMessage** (xsd:string) – contains more information in case of an error
- **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.

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 subscription ID is invalid or it does not belong to a transaction initiated by the same account

5.4. cancelSubscription

This method must be called if the user selects the option to cancel the subscription. It takes only one parameter:

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

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 that the user has to be redirected to to complete the cancellation. This is currently only required for Bouygues Telecom; on the other operators, the cancellation process is completed when this method returns. In that case, the return value will be an empty string.

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

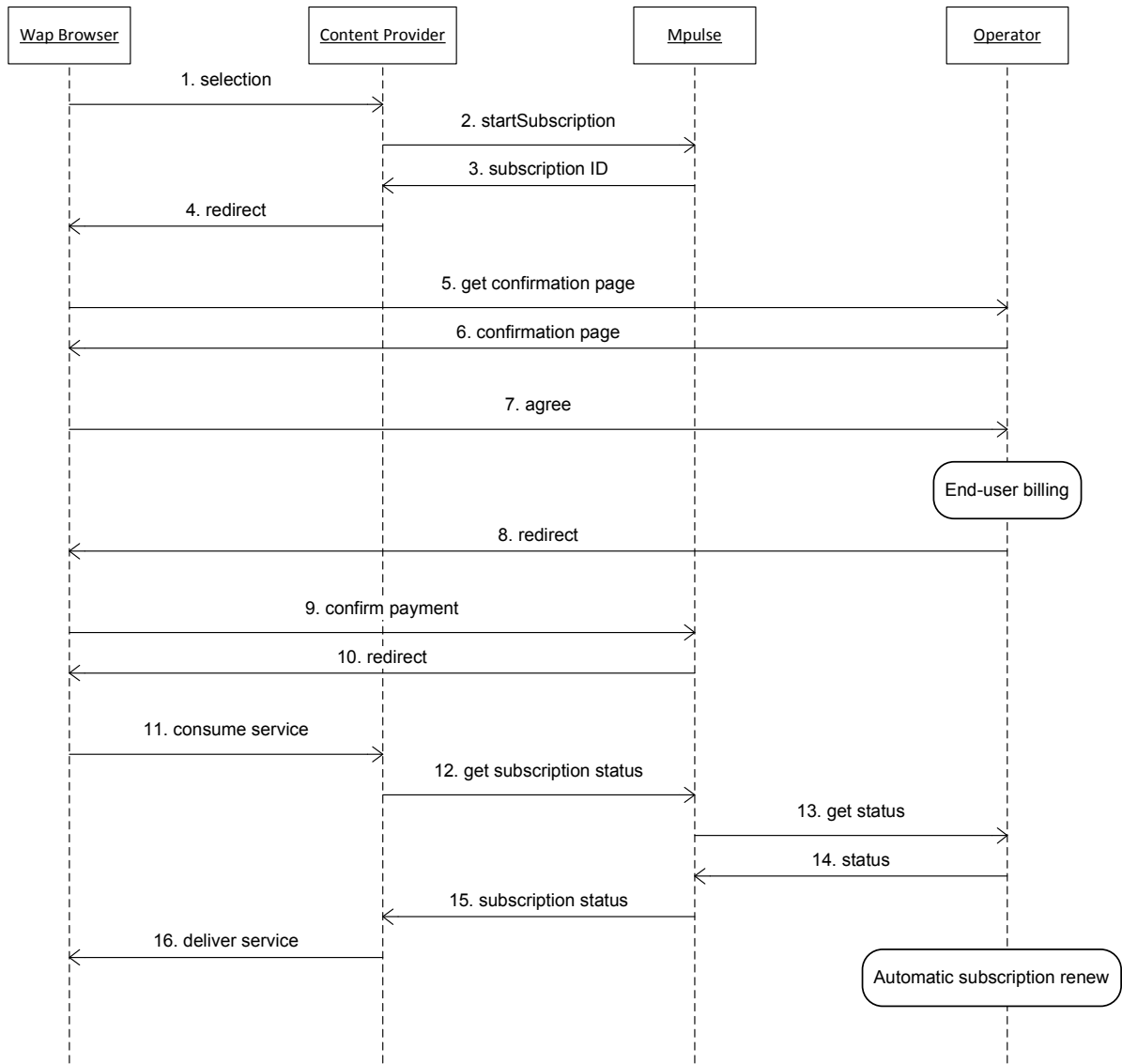
- **AuthenticationException** – occurs if no valid mobile-web payment account can be found with the credentials provided through the HTTP Basic Authentication mechanism or the account has not sufficient permissions to execute this method
- **BillingException** – occurs if the subscription ID is invalid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

5.5. 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.

In certain cases, the operators will take care of the automatic subscription renewals at the end of a period, so there is no explicit interaction between Mpulse and the operator and steps 13&14 will be skipped as they are implicit within the operator's systems.



6. Redirection

In order for the above scenarios to work, the mobile 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 Mpulse Gateway 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 WAP browser to redirected back to the content provider's WAP 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.

Depending on the workflows implemented by the operators, the end-user might have to be redirected to the operator's portal to confirm a transaction or a subscription. Mpulse will take care of the necessary workflow enhancements and redirections and makes sure all the necessary steps have been completed before the user is redirected to the content provider's portal.

7. Refunds

In order to increase customer satisfaction, operators require that an end-user is refunded for the value of a transaction if the download of the product could not be completed successfully. Several different methods can be used to detect whether a product has been completed:

1. For executable content, the Install Notify method should be used on terminals that support it to detect whether the item has been installed correctly
2. If 1 doesn't apply, the content provider must make sure that the last byte of the content item has successfully been downloaded
3. If the last byte has not been downloaded, allow 2 more attempts within 10 minutes for this content to be downloaded. Since a user cannot uniquely be identified by the content provider, this can lead to fraud when other users are simultaneously trying to download the same content. However, the risk can be minimized by limiting the download options to the user agent of the terminal (as provided in the HTTP headers) that initiated the transaction.

If an unsuccessful download has been detected, the end-user must be refunded for the amount of the transaction.

7.1. refundTransaction

If a download was unsuccessful, the end-user must be refunded for the amount paid. This only applies to single transactions and not to subscriptions. The method takes one parameter and has no return values:

- **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 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