B2C API Specification

# Approvals

This document needs below approvals for implementation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Author(s)** | **Name** | **Signature** | **Date** |
| Service Development Engineer | Eneth Kubai |  |  |
| **Reviewers** | **Name** | **Signature** | **Date** |
| Senior Manager Service Development | Reginald Tole |  |  |

# Revision Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Revision Date** | **Revision** | **Revision made by** |
| 1.0 | 18th June, 2013 | Initial Draft | Eneth Kubai |
| 1.1 | 9th July, 2013 | Added section 3.8.1: Password Encryption | Eric Mokaya |
| 1.2 | 29th July 2013 | Added Result codes | Eneth Kubai |
| 1.3 | 13th Jan 2014 | Added Result code explanations | Eneth Kubai |
| 1.4 | 7th Sept 2014 | Added changePassword interface, edited queryTransaction Interface, edited OriginatorConversationId field restrictions. Removed encrypted parameter tag. Edited the genericAPI request, response and result, added result and response codes | Eneth Kubai |
| 1.5 | 11th Nov 2014 | Attached SSL Guide | Eneth Kubai |

# References

|  |  |  |
| --- | --- | --- |
| **Document** | **Author** | **Name** |
|  | Eneth Kubai | Response and Result Codes V1.0 |
|  | John Barii | SSL Guide |

# Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| API | Application Programming Interface |
| B2C | Business to Customer |
| Broker | Service Access Gateway |
| SP | Service Provider |

Table of Contents

[Approvals ii](#_Toc402255545)

[Revision Log ii](#_Toc402255546)

[Abbreviations iv](#_Toc402255547)

[1 Introduction 7](#_Toc402255548)

[1.1 Scope 7](#_Toc402255549)

[2 Message Flow Description 8](#_Toc402255550)

[3 Data Type Definition 9](#_Toc402255551)

[3.1 IdentityType enumeration 9](#_Toc402255552)

[3.2 IdentifierType enumeration 9](#_Toc402255553)

[3.3 ParameterType structure 9](#_Toc402255554)

[3.4 Parameters structure 9](#_Toc402255555)

[3.5 ReferenceData structure 9](#_Toc402255556)

[3.6 Transaction structure 10](#_Toc402255557)

[3.7 Caller structure 10](#_Toc402255558)

[3.8 Initiator structure 11](#_Toc402255559)

[3.8.1 Password Encryption 11](#_Toc402255560)

[3.9 PrimaryParty structure 12](#_Toc402255561)

[3.10 ReceiverParty structure 12](#_Toc402255562)

[3.11 AccessDevice structure 12](#_Toc402255563)

[3.12 Identity structure 12](#_Toc402255564)

[3.13 Request structure 12](#_Toc402255565)

[3.14 Response structure 13](#_Toc402255566)

[3.15 ResultParameters structure 13](#_Toc402255567)

[3.16 Result structure 13](#_Toc402255568)

[3.17 Result code 14](#_Toc402255569)

[4 Web Service Interface Definition 20](#_Toc402255570)

[4.1 Interface: RequestMgrPortType 20](#_Toc402255571)

[4.1.1 Operation: GenericAPIRequest 20](#_Toc402255572)

[4.2 Interface: ResultMgrPortType 22](#_Toc402255573)

[4.2.1 Operation: GenericAPIResult 22](#_Toc402255574)

[4.3 Interface: QueueTimeoutNotificationPort 26](#_Toc402255575)

[4.3.1 Operation: notifyQueueTimeout 26](#_Toc402255576)

[4.4 Interface: QueryTransactionPort 27](#_Toc402255577)

[4.4.1 Operation: queryTransaction 27](#_Toc402255578)

[4.5 Interface: Management 29](#_Toc402255579)

[4.5.1 Operation: changePassword 29](#_Toc402255580)

[5 Example 33](#_Toc402255581)

[5.1 GenericAPIRequest 33](#_Toc402255582)

[5.1.1 The 3rd party send B2C request to the Broker 33](#_Toc402255583)

[5.1.2 The Broker return response to the 3rd party 38](#_Toc402255584)

[5.2 GenericAPIResult 40](#_Toc402255585)

[5.2.1 The Broker send notification to the 3rd party 40](#_Toc402255586)

[5.2.2 The 3rd party return response to the Broker 43](#_Toc402255587)

[5.3 notifyQueueTimeout 43](#_Toc402255588)

[5.3.1 The Broker send notification to the 3rd party 43](#_Toc402255589)

[5.3.2 The 3rd party return response to the Broker 45](#_Toc402255590)

[5.4 queryTransaction 46](#_Toc402255591)

[5.4.1 The 3rd party sends query request to the Broker 46](#_Toc402255592)

[5.4.2 The Broker return response to the 3rd party 47](#_Toc402255593)

[5.5 changePassword 50](#_Toc402255594)

[5.5.1 Request example: 50](#_Toc402255595)

[5.5.2 Response example: 51](#_Toc402255596)

# Introduction

## Scope

The present document specifies the real time B2C Web Service aspects of the interface. All aspects of B2C Web Service are defined here, these being:

* Message Flow Description
* Data Type Definition
* Web Service Interface Definition
* WSDL for this specification
* Example

# Message Flow Description



# Data Type Definition

## IdentityType enumeration

List of IdentityType values.

|  |  |
| --- | --- |
| Enumeration | Description |
| 1000 | Customer |
| 2000 | SPOperator |
| 3000 | OrganizationOperator |
| 5000 | Organization |
| 6000 | Till |
| 8000 | SP |

## IdentifierType enumeration

List of IdentityType values.

|  |  |
| --- | --- |
| Enumeration | Description |
| 1 | MSISDN |
| 2 | TillNumber |
| 3 | SPShortCode |
| 4 | OrganizationShortCode |
| 5 | IdentityID |
| 6 | O2CLink |
| 9 | SPOperatorCode |
| 10 | POSNumber |
| 11 | OrganizationOperatorUserName |
| 12 | OrganizationOperatorCode |
| 13 | VoucherCode |

## ParameterType structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| Key | xsd:string | No | It indicates a parameter name. |
| Value | xsd:string | No | It indicates a parameter value. |

## Parameters structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| Parameter | ParameterType[1..unbounded] | No | It is used to carry specific parameters for specific transaction or business operation. |

## ReferenceData structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| ReferenceItem | ParameterType[1..unbounded] | No | It is used carry some reference data that MM need not analyze but need to record it into transaction log.. |

## Transaction structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| CommandID | xsd:string | No | The unique identifier of transaction/business operation. Max length is 64.eg  *SalaryPayment*  *BusinessPayment*  *BusinessPaymentWithWithdrawalChargePaid*  *SalaryPaymentWithWithdrawalChargePaid*  *PromotionPayment*  *TransferFromBankToCustomer* |
| LanguageCode | xsd:string | Yes | It indicates language. It’s reserved. |
| OriginatorConversationID | xsd:string | No | The unique identifier of the request message generated by third party. It is used to identify a request between the third party and MM. Max length is 128.  Field must start with the B2C organisation short and name of organisation. Eg.  232323\_KCBOrg\_XXXXXX  XXXXX must be unique for every transaction. |
| ConversationID | xsd:string | Yes | The unique identifier generated by MM for a previous request message. It is used to support communication multi-times between the third party and MM for one operation/transaction. |
| Remark | xsd:string | Yes | The remark information about this operation. Max length is 255 |
| EncryptedParameters | xsd:string | Yes | It is used to carry the value for the element Parameters which are encrypted.  The value for this parameter should be a CDATA and encode with base64 |
| Parameters | Parameters | Yes | It is used to carry specific parameters for specific transaction or business operation. If the element EncryptedParameters presents, this parameter should not present. |
| ReferenceData | ReferenceData | Yes | It is used carry some reference data that MM need not analyze but need to record it into transaction log. |
| Timestamp | xsd:string | No | The timestamp generated by the third party. |

## Caller structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| CallerType | xsd:integer | No | Indicates the type of the caller: 2-APICaller 3-Other(Reserved) |
| ThirdPartyID | xsd:string | No | The unique identifier of a third party system defined in MM. It indicates the third party which initiates the request. Max length is 20 |
| Password | xsd:string | Yes | This security credential of the ThirdPartyID defined in MM. If the password feature for third party is used in MM, then this parameter must be presented in the request message. |
| CheckSum | xsd:string | Yes | Currently it is unused. It is reserved for API security. |
| ResultURL | xsd:string | Yes | It indicates the destination URL where Broker should send the result message to. |

## Initiator structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| IdentifierType | IdentifierType | No | It indicates the identifier type of the initiator. The value of this parameter must be a valid identifier type supported by MM. |
| Identifier | xsd:string | No | It indicates the identifier of the initiator. Its value must match the inputted value of the parameter IdentifierType. |
| SecurityCredential | xsd:string | No | It indicates the security credential of the initiator. Its value must match the inputted value of the parameter IdentifierType. |
| ShortCode | xsd:string | No | When the initiator is an organization operator, this parameter must be present in the request to indicate which organization the operator belongs to. If the initiator is not an organization operator, this parameter should not be present. |

### Password Encryption

The Caller will be required to confirm its authority to act on behalf of the Initiator (in other words, a specific B2C organisation) by presenting the user name and password for the Initiator, the latter encrypted with the public key from an X509 certificate issued to the Initiator specifically for this purpose.

The following algorithm must be followed by the Initiator to encrypt passwords:

First, create the block of data to be encrypted:

* Write the unencrypted password value.
* Then, encrypt the block of data created in step 1 with the public portion of the password key certificate. Use the RSA algorithm, and use PKCS #1.5 padding (not OAEP), and add the result to the encrypted stream – this becomes the encrypted password which is submitted via the API.
* Convert the resulting encrypted byte array into a string using base64 encoding. Present this base64 encoded string in the API request as the initiator SecurityCredential value.

## PrimaryParty structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| IdentifierType | IdentifierType | No | It indicates the identifier type of the primary party. The value of this parameter must be a valid identifier type supported by MM and must match the inputted value of the parameter IdentityType. |
| Identifier | xsd:string | No | It indicates a parameter value. |
| ShortCode | xsd:string | Yes | It is reserved |

## ReceiverParty structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| IdentifierType | IdentifierType | No | It indicates the identifier type of the recipient party. The value of this parameter must be a valid identifier type supported by MM. |
| Identifier | xsd:string | No | It indicates the identifier of the recipient party. Its value must match the inputted value of the parameter IdentifierType. |
| ShortCode | xsd:string | Yes | When the receiver party is an organization operator or a Till, this parameter must be present in the request to indicate which organization the receiver party belongs to. If the receiver party is not an organization operator or a Till, this parameter should not be present. |

## AccessDevice structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| IdentifierType | IdentifierType | No | It indicates the identifier type of the access device. |
| Identifier | xsd:string | No | It indicates the identifier of the access device. Its value must match the inputted value of parameter IdentifierType |

## Identity structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| Caller | Caller | No | It indicates the third party which initiates the request |
| Initiator | Initiator | No | It indicates the identity who makes the request |
| PrimaryParty | PrimaryParty | Yes | If business operation/action, this element is not present; if transaction, this can be either the debit party or the credit party according to the transaction type. |
| ReceiverParty | ReceiverParty | Yes | If business operation/action, this is the affected party; if transaction, it is the opposite party to the PrimaryParty |
| AccessDevice | AccessDevice | Yes | It indicates the access device which the initiator uses to initiate the request. |

## Request structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| Transaction | Transaction | No | It indicates a transaction. |
| Identity | Identity | No | This section is used to specify all identities involved in the request |
| KeyOwner | xsd:integer | No | It indicates which Key is used to encrypt the elements Initator.SecurityCredential and the EncryptedParameters. Its value are enumerated as follows: 1:the API Caller's Key 2:the Initiator's Key |

## Response structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| ResponseCode | xsd:string | No | It indicates whether MM accepts the request or not. |
| ResponseDesc | xsd:string | Yes | Its value is a description for the parameter ResultCode. |
| ConversationID | xsd:string | Yes | The unique identifier generated by M-Pesa for the request message. |
| OriginatorConversationID | xsd:string | Yes | The unique identifier generated by the third party for the request message. |
| ServiceStatus | xsd: integer | Yes | It indicates the MM service status. |

## ResultParameters structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| ResultParameter | ParameterType[0…unbounded] | Yes | It is used to carry specific parameters for specific transaction or business operation. |

## Result structure

|  |  |  |  |
| --- | --- | --- | --- |
| Element name | Element type | Optional | Description |
| ResultType | xsd:integer | Yes | 0: completed 1: waiting for further messages |
| ResultCode | xsd:string | No | It indicates whether MM processes the request successfully or not. Max length is 10 |
| ResultDesc | xsd:string | Yes | Its value is a description for the parameter ResultCode.Max length is 1024 |
| OriginatorConversationID | xsd:string | Yes | The unique identifier of the request message generated by third party. Its value comes from the request message. |
| ConversationID | xsd:string | Yes | The unique identifier generated by MM for a request |
| TransactionID | xsd:string | Yes | It’s only for transaction. When the request is a transaction request, MM will generate a unique identifier for the transaction. |
| ResultParameters | ResultParameters | Yes | It is used to carry specific parameters for specific transaction or business operation. |
| ReferenceData | ReferenceData | Yes | It comes from the request message |

## Result code

|  |  |  |
| --- | --- | --- |
| Error code | Error Description |  |
|  |  |  |
| 0 | Success | ApiResult |
| 1 | Insufficient Funds | ApiResult |
| 2 | Less Than Minimum Transaction Value | ApiResult |
| 3 | More Than Maximum Transaction Value | ApiResult |
| 4 | Would Exceed Daily Transfer Limit | ApiResult |
| 5 | Would Exceed Minimum Balance | ApiResult |
| 6 | Unresolved Primary Party | ApiResult |
| 7 | Unresolved Receiver Party | ApiResult |
| 8 | Would Exceed Maxiumum Balance | ApiResult |
| 11 | Debit Account Invalid | ApiResult |
| 12 | Credit Account Invaliud | ApiResult |
| 13 | Unresolved Debit Account | ApiResult |
| 14 | Unresolved Credit Account | ApiResult |
| 15 | Duplicate Detected | ApiResult |
| 17 | Internal Failure | ApiResult |
| 18 | Initiator Credential Check Failure | ApiResult |
| 19 | Message Sequencing Failure | ApiResult |
| 20 | Unresolved Initiator | ApiResult |
| 21 | Initiator to Primary Party Permission Failure | ApiResult |
| 22 | Initiator to Receiver Party Permission Failure | ApiResult |
| 23 | Request schema validation error | ApiResponse |
| 24 | MissingRequestParameters | ApiResponse |
| 25 | InvalidRequestParameters | ApiResponse |
| 26 | SystemTooBusy | ApiResponse |
| 0 | Success | ApiResponse |
| 100000000 | Request was cached, waiting for resending | ApiResponse |
| 100000001 | The system is overload | ApiResponse |
| 100000002 | Throttling error | ApiResponse |
| 100000003 | Exceed the limitation of the LICENSE | ApiResponse |
| 100000004 | Internal Server Error | ApiResponse |
| 100000005 | Invalid input value:%1  %1 indicates the parameter’s name. | ApiResponse |
| 100000006 | SP’s status is abnormal | ApiResponse |
| 100000007 | Authentication failed | ApiResponse |
| 100000008 | Service’s status is abnormal | ApiResponse |
| 100000009 | API’s status is abnormal | ApiResponse |
| 100000010 | Insufficient permissions | ApiResponse |
| 100000011 | Exceed the limitation of request rate | ApiResponse |
| 100000012 | Insufficient balance | ApiResponse |
| 100000013 | No route | ApiResponse |
| 100000014 | Missing mandatory parameter:%1  %1 indicates the parameter’s name. | ApiResponse |
| 28 | InitiatorAllowedOperationCheckFailure | ApiResult |
| 29 | InvalidCommand | ApiResult |
| 30 | ErrorSerializingRequest | ApiResponse |
| 31 | InitiatorNotSpecified | ApiResult |
| 32 | ErrorSerializingRequest | ApiResult |
| 33 | PrimaryPartyNotSpecified | ApiResult |
| 34 | PrimaryPartyIdentifierInvalid | ApiResult |
| 35 | ReceiverPartyNotSpecified | ApiResult |
| 36 | ReceiverPartyIdentifierInvalid | ApiResult |
| 37 | MissingApiCommand | ApiResult |
| 38 | InvalidConversationId | ApiResult |
| 39 | UnknownConversationId | ApiResult |
| 40 | InvalidParameterDefinition | ApiResult |
| 41 | DuplicateConversationDetected | ApiResult |
| 42 | DuplicateStageMessageDetected | ApiResult |
| 43 | AwaitingConfirmation | ApiResult |
| 44 | InitiatorToReceiverPartyPermissionFailure | ApiResult |
| 45 | InternalErrorDuringFinancialTransaction | ApiResult |
| 46 | ConfirmationReceived | ApiResult |
| 47 | RejectionReceived | ApiResult |
| 48 | OperationPermissionFailure | ApiResult |
| 49 | NoTransactionFound | ApiResult |
| 50 | InitiatorOrgIdentityStatusFailure | ApiResult |
| 51 | DebitChargeAccountInvalid | ApiResult |
| 52 | WouldExceedMaximumSingleAirtimePurchase | ApiResult |

The following table lists result codes and result descriptions which are provided to the caller. These may be commincated either in the synchronous *ApiResponse* message, or in the asynchronous *ApiResult* message.

1. **Success**

This is indicated for both ApiResponse and ApiResult messages. An ApiResponse value of 0 indicates that the request has passed basic validation tests and been passed on to the core system for further processing. An ApiResult value of 0 means that the requested financial transaction has been completed successfully. In case of multi staged transaction the response code will be 43 for initiation stage and 46 for confirmation.

1. **InsufficientFunds**

Typically indicates that the primary party does not have enough money to complete the requested financial transaction. In the case of the B2C API, this means that the business organisation does not have enough money in its utility account.

This is returned by the *ApiResult* and represented internally as a transaction reason.

1. **LessThanMinimumTransactionValue**
2. **MoreThanMaximumTransactionValue**

Each of these potential failures derive from the rules which govern MPesa financial transactions. In the case of the B2C API, these are taken from *DefaultBusinessUtilityAccount* rules, which specify a lower bound of 10 KE shillings and an upper bound of 70000. For promotion payments and salary payments to unregistered users, the upper limit is lower: 35000. Likewise, the lower bound for these 2 transactions is also different: 101 rather than 10.

These are both returned by the *ApiResult* and represented internally as a transaction reason.

1. **WouldExceedDailyTransferLimit**

This is a limit on daily activity. For the business organisation, this is very high: 100000000 KE shillings. It is much more likely that this rule will apply to the customer, where the limit is 140000 KE shillings.

This is returned by the *ApiResult* and represented internally as a transaction reason.

1. **WouldExceedMinimumBalance**

This rule is rather confusingly named – it actually means that a transaction would bring the business organisations utility account bellow the required minimum – which is currently 0.

This is returned by the *ApiResult* and represented internally as a transaction reason. It is computed during execution of the financial transaction.

1. **UnresolvedPrimaryParty**
2. **UnresolvedReceiverParty**

The primary party or receiver party cannot be associated with an MPesa identity. For transactions (such as promotion payment and salary payment) which are supported for unregistered users, the unresolved receiver party rule is not applied.

These are returned in the *ApiResult* and checked before the linked financial transaction is created.

1. **WouldExceedMaximumBalance**

For the B2C API, this applies to the recipient MMF customer – the limit here is 100,000 KE shillings.

This is returned by the *ApiResult* and represented internally as a transaction reason. It is computed during execution of the financial transaction.

1. **LessThanMinAirtimeValue**
2. **MoreThanMaxAirtimeValue**

Neither of these apply to B2C API operations.

1. **DebitAccountInvalid**
2. **CreditAccountInvalid**
3. **UnresolvedDebitAccount**
4. **UnresolvedCreditAccount**

These are all existing failure transaction reasons for B2C financial transactions – they would be issued to the caller in the *ApiResult* message. The latter 2 in particular are unlikely to occur, as the request would be rejected by Core API specific checks before the request reached the accounting engine.

1. **DuplicateDetected**

Currently, any requests which presents an originator conversation id which has been seen before will be rejected. This rule will need to be reconsidered in contexts which require multi-stage conversations or multiple Callers. This check (along with the message expiry check) is the first thing done by the transaction processor when handling an API request. The outcome is reported in the *ApiResult* message.

1. **PayUtilityInvalidAccountNumberFormat**

Does not apply to the B2C API operations.

1. **InternalFailure**

A catch all for failures which are not identified more specifically – this can occur in either the *ApiResponse* or *ApiResult* – although the intent is to replace any such error with a more precise message.

1. **InitiatorCredentialCheckFailure**

The password check for the initiator failed, either because the presented password is wrong, or something has gone wrong in the encryption or decryption steps. This is issued in the *ApiResult* before the creation of a financial transaction.

1. **ApiRequestMessageExpiryFailure**

Initiators have the option to specify a request timestamp. If the gap between this value and the time at which the message is received by the transaction processor is too great, then the request is rejected. The specific interval is configured separately for each API operation (and indeed, for each API operation stage). This is the first check performed on an API request arriving at the transaction processor, and as such is included in the *ApiResult* message.

1. **UnresolvedInitiator**

The initiator username presented with the request cannot be found. This is included in the *ApiResult*.

1. **InitiatorToPrimaryPartyPermissionFailure**

The initiator presented does not have the right to issue requests for the specified primary party (as established during creation of the initiator on the admin web site). The is part of the *ApiResult*, and computed prior to issuing a financial transaction.

1. **InitiatorStatusCheckFailure**

The presented initiator username can be received, but the initiator is not currently active. This fact is returned in the *ApiResult*.

1. **RequestSchemaValidationError**

Incoming API requests are validated against the schema defined in *CPSInterface\_Request.xsd*. This is part of the *ApiResponse* message – the details of the validation error are included in the response.

1. **MissingRequestParameters**

Required input parameters are defined for each type of API operation. If these are missing, then this is issued in the *ApiResult* message. The names of the missing parameters are included in the result parameters field.

1. **InvalidRequestParameters**

If all required parameters are presented, then validation checks are performed. Specifically, the parameter is checked to see if it can be converted to the intended type, and then checked against (optional) configuration regular expression based validation rules. For the B2C API, only the type check (i.e the amount specified is a valid decimal) is performed. Failures are indicated in the *ApiResult* message, with the details included in the result parameters.

1. **SystemTooBusy**

Included in the *ApiResponse* message if a traffic blocking condition is in place. See section 5.1 for more detail.

1. **InitiatorAllowedOperationCheckFailure**

Upon creation, initiators are assigned permissions for specific API operations – if the operation specified in the request message is not included in this list, then this *ApiResult* message is received.

1. **InvalidCommand**

The command specified in the request is not defined – this is part of the *ApiResult*.

1. **ErrorSerializingRequest**

After XML schema validation, the API attempts to convert the XML request into an internal *ApiRequest* object – any failures result in this *ApiResponse* message. The details of the exception are included.

1. **InitiatorNotSpecified**

The initiator username is not specified, or not specified in a way that can be parsed.

1. **ErrorSerializingRequest**

The presented identifier for the initiator is not a username.

1. **PrimaryPartyNotSpecified**
2. **PrimaryPartyIdentifierInvalid**
3. **ReceiverPartyNotSpecified**
4. **ReceiverPartyIdentifierInvalid**

In all these case, one of the API parties is either missing from the request, or has been presented with the wrong identifier type., This kind of failure is indicated in the *ApiResult* message.,

1. **MissingApiCommand**

No command is included in the request – in general this problem should be captured by the XML schema validation, but if it is not this *ApiResult* message is issued.

1. **InvalidConversationId**
2. **UnknownConversationId**

Not currently checked for – this will take place for multi-stage conversations.

1. **InvalidParameterDefinition**

Not been used.

1. **DuplicateConversationDetected**

If the originatorconversationid is be reused again, the request will end up with duplicate error code. The request should have a unique originatorconversationid for every new apirequest.

1. **DuplicateStageMessageDetected**

If the stage name is requested twice it will end up with duplicate stage. Though the occurance of this would be very rare.

1. **AwaitingConfirmation**

It is applicable for multi stage transaction. Once the first stage of the transaction is set to authorised with will set the result code as AwaitingConfirmation to convey the initiator to initiate the confirmation stage.

1. **InitiatorToReceiverPartyPermissionFailure**

The initiator presented does not have the right to issue requests for the specified receiver party (as established during creation of the initiator on the admin web site). The is part of the *ApiResult*, and computed prior to issuing a financial transaction.

1. **InternalErrorDuringFinancialTransaction**
2. **ConfirmationReceived**

It is applicable for the multi stage transaction. The result code is send once the confirmation stage is processed successfully to confirm the financial transaction.

1. **RejectionReceived**

It is applicable for the multi stage transaction. The result code is send once the confirmation stage is processed successfully to cancel the financial transaction

1. **OperationPermissionFailure**

For transaction status query if the initiator presented was not the part of the original transaction or linked to the respective parties within the original request a transaction access permission failure result is send out. For balance query if the initiator is not linked to the request organisation shortcode it will return the same result code. As the validation is common the result code will be shared by both the request.

1. **NoTransactionFound**

For transaction status query is the request transaction parameter is not found in the system it will send the transaction not found result code.

1. **InitiatorOrgIdentityStatusFailure**

The result code is returned if the organisation status is closed.

1. **DebitChargeAccountInvalid**

If the Airtime purchase account is set to inactive state the corresponding transaction will return an invalid debit charge account failure.

1. **WouldExceedMaximumSingleAirtimePurchase**

If transaction amount exceeds the maximum amount configured for single request it will fail the transaction for rule failure.

# Web Service Interface Definition

## Interface: RequestMgrPortType

### Operation: GenericAPIRequest

The 3rd party invokes this operation to send a B2C request

#### 4.1.1.1 Message Header: RequestSOAPHeader

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| spId | xsd: string | No | SP ID.  This is the Service Provider Identifier that is allocated by the Broker to the 3rd party.  [Example] 000201 |
| spPassword | xsd: string | Yes | This is an encrypted form of the SP password issued to an SP when an account is created on the Broker.  The encrypted password is a Base64 encoded string of the SHA-256 hash of the concatenation of the spId, password and the timeStamp as illustrated below:  Given the following parameters  spId: 601399  password: spPassword  timestamp: 20130702212854  spPassword = BASE64(SHA-256(spId + Password + timeStamp)) e.g.  spPassword = BASE64(SHA-256(601399spPassword20130702212854)  [Example]  e6434ef249df55c7a21a0b45758a39bb |
| serviceId | xsd: string | Yes | Service ID.  This is the Service Identifier that is allocated by the Broker for every service created.  [Example]  3500001000012 |
| Timestamp | xsd: string | Yes | Time stamp (UTC time).  The value is required during SHA-256 encryption for **spPassword**.  NOTE  If the **spPassword** parameter must be set, this parameter is mandatory.  [Format]  yyyyMMddHHmmss  [Example]  20100731064245 |

#### 4.1.1.2 Input Message: RequestMsg

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| RequestMsg | xsd: string | No | Request Message from 3rd party. Its value should be an instance of Request Type and a CDATA |

**Note:**

1. If there is no configuration for notification URL on Broker side, which indicates the callback url for accepting notification of GenericAPIResult, the **ResultURL** parameter inside Identity tag must present.

2. If there is no configuration for notification URL on Broker side, which indicates the callback url for accepting notification of cached requests expired, the 3rd party must add a key-pair parameter into **ReferenceData** andthe key is **QueueTimeoutURL**.

#### 4.1.1.3 Output Message: ResponseMsg

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| ResponseMsg | xsd: string | No | Response return to 3rd party. Its value should be an instance of Response Type and a CDATA. |

## Interface: ResultMgrPortType

### Operation: GenericAPIResult

This operation must be implemented by a Web Service at the 3rd party side if it requires notification of the final result for B2C request. It will be invoked by Broker to notify the 3rd party once Broker received the notification from CoreAPI.

#### 4.2.1.1 Input Message: ResultMsg

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| ResultMsg | xsd: string | No | Request Message from Broker. Its value should be a instance of Result Type and a CDATA. |

#### 4.2.1.2 Output Message: ResponseMsg

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| ResponseMsg | xsd: string | No | Response return to Broker. Its value should be a instance of Response Type and a CDATA. |

#### 4.2.1.3 Response Codes

|  |  |
| --- | --- |
| ResponseCode | ResponseDesc |
| 000000000 | Success |
| 200000001 | The system is overload |
| 200000002 | Throttling error |
| 200000003 | Internal Server Error |

The parameters in the **Result/ResultParameters** element are listed below:

The **ResultParameters** will change to this on April 2015.

| Parameter | Data Type | Mandatory or Optional | Description | Example |
| --- | --- | --- | --- | --- |
| TransactionReceipt | xs:string | Mandatory | Unique transaction ID for the payment transaction. | 1234560000007031 |
| TransactionAmount | xs:string | Mandatory | 2 fixed point decimal amount of the transaction | 100.22 |
| TransactionCompletedDateTime | xs:string | Mandatory | The time when the financial transaction was completed, as recorded in the M-Pesa system.  Format:  dd.mm.yyyy HH:MM:SS | 22.03.2012 12:20:20 |
| ReceiverPartyPublicName | xs:string | Mandatory | Public name of the customer who has been issued the money.  Format: *<MSISDN>-<First Name> <Last Name>*  For unregister customer, the *<First Name> <Last Name>* will be blank. | 254778866553868-Kip Keino |
| B2CWorkingAccountAvailableFunds | xs:string | Mandatory | 2 fixed point decimal amount of the Available Balance of the organization’s Working Account.  Note: the exact Account Type will depend on the configuration. | 9999.88 |
| B2CUtilityAccountAvailableFunds | xs:string | Mandatory | 2 fixed point decimal amount of the Available Balance of the organization’s Utility Account, which is the debit account of the principle transaction.  Note: the exact Account Type will depend on the configuration. | 9999.88 |
| B2CChargesPaidAccountAvailableFunds | xs:string | Mandatory | 2 fixed point decimal amount of the Available Balance of the organization’s Charge Paid Account, which is debit account of the charge.  Note: the exact Account Type will depend on the configuration. | 9999.88 |
| B2CRecipientIsRegisteredCustomer | xs:string | Mandatory | Indicate if the recipient is an existing Mobile Money customer, or an unregistered customer.  Format: Y or N  Note: This field is not filled by the G1 platform, but it will be filled by the G2 platform. | Y |

## Interface: QueueTimeoutNotificationPort

### Operation: notifyQueueTimeout

This operation must be implemented by a Web Service at the 3rd party side if it requires notification of cached B2C requests are expired. It will be invoked by Broker to notify the 3rd party once cached B2C requests are expired.

#### 4.3.1.1 Input Message: notifyQueueTimeout

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| originatorConversationID | xsd:string |  | originatorConversationID from the request sent by the 3rd party |
| originRequest | xsd:string | No | Original request without SOAP Header sent by 3rd party. Its value is encoded with base64, when the 3rd party receive the request, it should decode it. |
| extensionInfo | Parameters | Yes | Extended parameters. |

#### 4.3.1.2 Output Message: notifyQueueTimeoutResponse

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| result | Result | No |  |
| extensionInfo | Parameters | Yes | Extended parameters. |

#### 4.3.1.3 Response Code

|  |  |
| --- | --- |
| ResponseCode | ResponseDesc |
| 000000000 | Success |
| 000000001 | Failed |

## Interface: QueryTransactionPort

### Operation: queryTransaction

The 3rd party invokes this operation to query transaction information..

#### 4.3.1.1 Message Header: RequestSOAPHeader

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| spId | xsd: string | No | SP ID.  It’s allocated by the Broker to the 3rd party.  [Example]  000201 |
| spPassword | xsd: string | Yes | Encrypted authentication password for partners to access the Broker.  The value is a character string encrypted from **spId** + **Password** + **timeStamp** by SHA-256. The encryption formula is as follows: spPassword =BASE64(SHA-256(spId + Password + timeStamp))  In the preceding formula:   * **timeStamp**: value of **timeStamp**. * **Password**: authentication password for 3rd parties to access the Broker. The value is allocated by the Broker.   NOTE  The authentication modes include SPID&Password, SPID&IP&Password, and SPID&IP. When the authentication mode is SPID&Password or SPID&IP&Password, this parameter is mandatory.  [Example]  e6434ef249df55c7a21a0b45758a39bb |
| serviceId | xsd: string | Yes | Service ID.  The value is allocated by the Broker to the 3rd party.  [Example]  3500001000012 |
| timeStamp | xsd: string | Yes | Time stamp (UTC time).  The value is required during SHA-256 encryption for **spPassword**.  NOTE  If the **spPassword** parameter must be set, this parameter is mandatory.  [Format]  yyyyMMddHHmmss  [Example]  20100731064245 |

#### 4.3.1.2 Input Message: queryTransaction

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| originatorConversationID | xsd:string |  | The unique identifier of the request message generated by third party. It is used to identify a request between the third party and MM. Max length is 128 |
| extensionInfo | Parameters | Yes | Extended parameters. |

extensionInfo Description

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Optional | Type | Description |
| queryDate | Yes | String(20) | The date of the original conversation. Format is yyyyMMddHHmmss, for example: 20131230134412  Note:  If this parameter does not present, it will cost more time to get the result. |

#### 4.3.1.3 Output Message: queryTransactionResponse

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| result | Response | No |  |
| submitApiRequestList | xsd:string[0-unbounded] | Y | Requests sent by the 3rd party. Its value is the requests sent by the 3rd party with base64 encoded. |
| submitApiResponseList | xsd:string[0-unbounded] | Y | Responses returned from the Broker. Its value is the responses returned from the Broker with base64 encoded. |
| submitApiResultList | xsd:string[0-unbounded] | Y | Results sent to the 3rd party. Its value is the requests sent by the Broker with base64 encoded. |
| queueTimeOutList | xsd:string[0-unbounded] | Y | QueueTimeout requests sent to the 3rd party. Its value is the requests sent by the Broker with base64 encoded. |
| extensionInfo | Parameters | Yes | Extended parameters. |

#### 4.3.1.4 Response Codes

|  |  |
| --- | --- |
| ResponseCode | ResponseDesc |
| 000000000 | Success |
| 100000001 | The system is overload |
| 100000002 | Throttling error |
| 100000003 | Exceed the limitation of the LICENSE |
| 100000004 | Internal Server Error |
| 100000005 | Invalid input value:%1  %1 indicates the parameter’s name. |
| 100000006 | SP’s status is abnormal |
| 100000007 | Authentication failed |
| 100000008 | Service’s status is abnormal |
| 100000010 | Insufficient permissions |
| 100000014 | Missing mandatory parameter:%1  %1 indicates the parameter’s name. |

## 4.5 Interface: Management

### 4.5.1 Operation: changePassword

The 3rd party invokes this operation to change his password.

* + - 1. Input Message: changePassword

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| spId | xsd: string | No | SP ID.  It’s allocated by the SAG to the 3rd party.  [Example]  000201 |
| spPassword | xsd: string | Yes | Encrypted authentication password for partners to access the SAG.  The value is a character string encrypted from **spId** + **Password** + **timeStamp** by SHA-256. The encryption formula is as follows: spPassword =BASE64(SHA-256(spId + Password + timeStamp))  In the preceding formula:  **timeStamp**: value of **timeStamp**.  **Password**: authentication password for 3rd parties to access the SAG. The value is allocated by the SAG.  NOTE  The authentication modes include SPID&Password, SPID&IP&Password, and SPID&IP. When the authentication mode is SPID&Password or SPID&IP&Password, this parameter is mandatory.  [Example]  e6434ef249df55c7a21a0b45758a39bb |
| timeStamp | xsd: string | Yes | Time stamp (UTC time).  The value is required during SHA-256 encryption for **spPassword**.  NOTE  If the **spPassword** parameter must be set, this parameter is mandatory.  [Format]  yyyyMMddHHmmss  [Example]  20100731064245 |
| newPassword | xsd:string | No | New authentication password for 3rd parties to access the SAG. It should be encrypted by AES-256 and encoded with base64. Shared key and will be allocated by the SAG.  For example:  New password is !QAZ2wsx,  Security key is AAAabcdefghijklm,  Vector is abcdefghijklmnop  BASE64(AES(!QAZ2wsx, AAAabcdefghijklm, abcdefghijklmnop)) is wi2a7BAH0QPd2LRdmcgC9w==  SP should fill wi2a7BAH0QPd2LRdmcgC9w== as newPassword |
| extensionInfo | Parameters | Yes | Extended parameters. |

* + - 1. Output Message: changePasswordResponse

| Element name | Element type | Optional | Description |
| --- | --- | --- | --- |
| result | Result | No | Result. |
| extensionInfo | Parameters | Yes | Extended parameters. |

* + - 1. Response Code

| ResponseCode | ResponseDesc |
| --- | --- |
| 000000000 | Success |
| 100000001 | The system is overload |
| 100000002 | Throttling error |
| 100000003 | Exceed the limitation of the LICENSE |
| 100000004 | Internal Server Error |
| 100000005 | Invalid input value:%1  %1 indicates the parameter’s name. |
| 100000006 | SP’s status is abnormal |
| 100000007 | Authentication failed |
| 100000014 | Missing mandatory parameter:%1  %1 indicates the parameter’s name. |

# Example

## GenericAPIRequest

### The 3rd party send B2C request to the Broker

Example1: Request with ResultURL and QueueTimeoutURL

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:req="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<soapenv:Header>

<tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2\_1">

<tns:spId>35000011</tns:spId>

<tns:spPassword>c5216e519a071d601bedd150f3fcd026</tns:spPassword>

<tns:serviceId>35000001000009</tns:serviceId>

<tns:timeStamp>20120101010101</tns:timeStamp>

</tns:RequestSOAPHeader>

</soapenv:Header>

<soapenv:Body>

<req:RequestMsg><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<request xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<Transaction>

<CommandID>PromotionPayment</CommandID>

<LanguageCode></LanguageCode>

<OriginatorConversationID>4596</OriginatorConversationID>

<ConversationID> </ConversationID>

<Remark> </Remark>

<Parameters>

<Parameter>

<Key>Amount</Key>

<Value>1</Value>

</Parameter>

<Parameter>

<Key>Key1</Key>

<Value>Value1</Value>

</Parameter>

</Parameters>

<ReferenceData>

<ReferenceItem>

<Key>QueueTimeoutURL</Key>

<Value>http://10.66.49.789:7888/new</Value>

</ReferenceItem>

<ReferenceItem>

<Key>Occasion</Key>

<Value>Jamuhuri</Value>

</ReferenceItem>

</ReferenceData>

<Timestamp>2013-07-29T18:50:41.2109675Z</Timestamp>

</Transaction>

<Identity>

<Caller>

<CallerType>2</CallerType>

<ThirdPartyID> </ThirdPartyID>

<Password>Password0</Password>

<CheckSum>CheckSum0</CheckSum>

<ResultURL>ResultURL0</ResultURL>

</Caller>

<Initiator>

<IdentifierType>11</IdentifierType>

<Identifier>username</Identifier>

<SecurityCredential>SecurityCredential0</SecurityCredential>

<ShortCode>859636</ShortCode>

</Initiator>

<PrimaryParty>

<IdentifierType>4</IdentifierType>

<Identifier>859636</Identifier>

<ShortCode>859636</ShortCode>

</PrimaryParty>

<ReceiverParty>

<IdentifierType>1</IdentifierType>

<Identifier>2547204789659</Identifier>

<ShortCode>ShortCode1</ShortCode>

</ReceiverParty>

<AccessDevice>

<IdentifierType>1</IdentifierType>

<Identifier>Identifier3</Identifier>

</AccessDevice>

</Identity>

<KeyOwner>1</KeyOwner>

</request>]]></req:RequestMsg>

</soapenv:Body>

</soapenv:Envelope>

Example2: Request without ResultURL and QueueTimeoutURL

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:req="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<soapenv:Header>

<tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com/schema/osg/common/v2\_1">

<tns:spId>35000001</tns:spId>

<tns:spPassword>c5216e519a071d601bedd150f3fcd026</tns:spPassword>

<tns:timeStamp>20080101010101</tns:timeStamp>

<tns:serviceId>35000001000009</tns:serviceId>

<tns:OA>861234567890</tns:OA>

<tns:FA>861234567890</tns:FA>

</tns:RequestSOAPHeader>

</soapenv:Header>

<soapenv:Body>

<req:RequestMsg><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<request xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<Transaction>

<CommandID>CommandID0</CommandID>

<LanguageCode>LanguageCode0</LanguageCode>

<OriginatorConversationID>OriginatorConversationID0</OriginatorConversationID>

<ConversationID>ConversationID0</ConversationID>

<Remark>Remark0</Remark>

<EncryptedParameters>EncryptedParameters0</EncryptedParameters>

<Parameters>

<Parameter>

<Key>Key0</Key>

<Value>Value0</Value>

</Parameter>

<Parameter>

<Key>Key1</Key>

<Value>Value1</Value>

</Parameter>

</Parameters>

<ReferenceData>

<ReferenceItem>

<Key>Key2</Key>

<Value>Value2</Value>

</ReferenceItem>

</ReferenceData>

<Timestamp>

</Timestamp>

</Transaction>

<Identity>

<Caller>

<CallerType>0</CallerType>

<ThirdPartyID>ThirdPartyID0</ThirdPartyID>

<Password>Password0</Password>

<CheckSum>CheckSum0</CheckSum>

</Caller>

<Initiator>

<IdentifierType>1</IdentifierType>

<Identifier>Identifier0</Identifier>

<SecurityCredential>SecurityCredential0</SecurityCredential>

<ShortCode>

</ShortCode>

</Initiator>

<PrimartyParty>

<IdentifierType>1</IdentifierType>

<Identifier>Identifier1</Identifier>

<ShortCode>ShortCode0</ShortCode>

</PrimartyParty>

<ReceiverParty>

<IdentifierType>1</IdentifierType>

<Identifier>Identifier2</Identifier>

<ShortCode>ShortCode1</ShortCode>

</ReceiverParty>

<AccessDevice>

<IdentifierType>1</IdentifierType>

<Identifier>Identifier3</Identifier>

</AccessDevice>

</Identity>

<KeyOwner>0</KeyOwner>

</request>]]></req:RequestMsg>

</soapenv:Body>

</soapenv:Envelope>

### The Broker return response to the 3rd party

Example1: Forward CoreAPI response to the 3rd party

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:req="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<soapenv:Header/>

<soapenv:Body>

<req:ResponseMsg><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<response xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/response">

<ResponseCode>ResponseCode0</ResponseCode>

<ResponseDesc>ResponseDesc0</ResponseDesc>

<ConversationID>

</ConversationID>

<OriginatorConversationID>

</OriginatorConversationID>

<ServiceStatus>0</ServiceStatus>

</response>]]></req:ResponseMsg>

</soapenv:Body>

</soapenv:Envelope>

Or

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Body>

<req:ResponseMsg xmlns:req="http://api-v1.gen.mm.vodafone.com/mminterface/request"><![CDATA[<?xml version="1.0" encoding="UTF-8"?><Response><ResponseCode>0</ResponseCode><ConversationID>AG\_20140825\_000056ac18ccfd6a13a2</ConversationID><ResponseDesc>Accept the service request successfully.</ResponseDesc><OriginatorConversationID>B2C-SIT-000005

</OriginatorConversationID><ServiceStatus>0</ServiceStatus></Response>]]></req:ResponseMsg>

</soapenv:Body>

</soapenv:Envelope>

Example2: Error response caused by authentication failed.

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:req="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<soapenv:Header/>

<soapenv:Body>

<req:ResponseMsg><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<response xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/response">

<ResponseCode>100000007</ResponseCode>

<ResponseDesc>Authentication failed</ResponseDesc>

</response>]]></req:ResponseMsg>

</soapenv:Body>

</soapenv:Envelope>

Example3: Error response caused by waiting for resending.

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:req="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<soapenv:Header/>

<soapenv:Body>

<req:ResponseMsg><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<response xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/response">

<ResponseCode>100000000</ResponseCode>

<ResponseDesc>Request was cached, waiting for resending</ResponseDesc>

</response>]]></req:ResponseMsg>

</soapenv:Body>

</soapenv:Envelope>

## GenericAPIResult

### The Broker send notification to the 3rd party

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<ResultMsg xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/result">

<![CDATA[<Result xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/result">

<ResultType>Completed</ResultType>

<ResultCode>0</ResultCode>

<ResultDesc>Success</ResultDesc>

<OriginatorConversationID>R9-2-eq-a7cb-q277</OriginatorConversationID>

<ConversationID>61da40ad-7e96-4889-83c4-1ea77091f8ec</ConversationID>

<TransactionID>X-IO213</TransactionID><ResultParameters>

<Parameter>

<Key xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">Amount</Key>

<Value xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">Ksh200.00</Value>

</Parameter><Parameter><Key xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">Transaction Datetime</Key>

<Value xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">22/10/2014 20:55:24</Value>

</Parameter><Parameter>

<Key xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">Working Account Available Funds</Key>

<Value xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">2338002.5100</Value>

</Parameter><Parameter>

<Key xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">Charges Paid Account Available Funds</Key>

<Value xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">0.0000</Value>

</Parameter><Parameter>

<Key xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">Utility Account Available Funds</Key>

<Value xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">461898.0000</Value>

</Parameter></ResultParameters>

<ReferenceData><ReferenceItem><Key xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">QueueTimeoutURL</Key><Value xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/request">http://10.66.49.201:8097/</Value></ReferenceItem></ReferenceData></Result>]]></ResultMsg>

</s:Body>

</s:Envelope>

Or

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Header/>

<soapenv:Body>

<req:ResultMsg xmlns:req="http://cps.huawei.com/cpsinterface/result"><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<Result>

<ResultType>0</ResultType>

<ResultCode>0</ResultCode>

<ResultDesc>The service request is processed successfully.</ResultDesc>

<OriginatorConversationID>D7866050-CE7C-46B9-BE97-D9FBACC69041</OriginatorConversationID>

<ConversationId>B72AC018-86E4-477A-B4B0-9EC65AE8A2FE</ConversationId>

<TransactionId>2782728972</TransactionId>

<ResultParameters>

<ResultParameter>

<Key>TransactionReceipt</Key>

<Value>2782728972</Value>

</ResultParameter>

<ResultParameter>

<Key>TransactionAmount</Key>

<Value>100.00</Value>

</ResultParameter>

<ResultParameter>

<Key>TransactionCompletedDateTime</Key>

<Value>22.03.2012 12:20:20</Value>

</ResultParameter>

<ResultParameter>

<Key>ReceiverPartyPublicName</Key>

<Value>254778866553868-Kip Keino</Value>

</ResultParameter>

<ResultParameter>

<Key>B2CWorkingAccountAvailableFunds</Key>

<Value>10000.00</Value>

</ResultParameter>

<ResultParameter>

<Key>B2CUtilityAccountAvailableFunds</Key>

<Value>100000.00</Value>

</ResultParameter>

<ResultParameter>

<Key>B2CChargesPaidAccountAvailableFunds</Key>

<Value>1000.00</Value>

</ResultParameter>

<ResultParameter>

<Key>B2CRecipientIsRegisteredCustomer</Key>

<Value>Y</Value>

</ResultParameter>

<ResultParameter>

<Key>B2CWithdrawalChargeAmount</Key>

<Value>12.00</Value>

</ResultParameter>

</ResultParameters>

<ReferenceData>

<ReferenceItem>

<Key>Occasion</Key>

<Value>Christmas</Value>

</ReferenceItem>

</ReferenceData>

</Result>]]></req:ResultMsg>

</soapenv:Body>

</soapenv:Envelope>

### The 3rd party return response to the Broker

Example1: Success response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:req="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<soapenv:Header/>

<soapenv:Body>

<req:ResponseMsg><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<response xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/response">

<ResponseCode>00000000</ResponseCode>

<ResponseDesc>success</ResponseDesc>

</response>]]></req:ResponseMsg>

</soapenv:Body>

</soapenv:Envelope>

Example2: Error response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:req="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<soapenv:Header/>

<soapenv:Body>

<req:ResponseMsg><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<response xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/response">

<ResponseCode>20000003</ResponseCode>

<ResponseDesc>Internal Server Error</ResponseDesc>

</response>]]></req:ResponseMsg>

</soapenv:Body>

</soapenv:Envelope>

## notifyQueueTimeout

### The Broker send notification to the 3rd party

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org/schema/timeoutnotification/data/v1\_0/local" xmlns:res="http://api-v1.gen.mm.vodafone.com/mminterface/result">

<soapenv:Header/>

<soapenv:Body>

<loc:notifyQueueTimeout>

<loc:originatorConversationID>OriginatorConversationID0</loc:originatorConversationID>

<loc:originRequest>PHNvYXBlbnY6RW52ZWxvcGUgeG1sbnM6c29hcGVudj0iaHR0cDovL3NjaGVtYXMueG1sc29hcC5v

cmcvc29hcC9lbnZlbG9wZS8iIHhtbG5zOnJlcT0iaHR0cDovL2FwaS12MS5nZW4ubW0udm9kYWZv

bmUuY29tL21taW50ZXJmYWNlL3JlcXVlc3QiPgogICA8c29hcGVudjpCb2R5PgogICAgICA8cmVx

OlJlcXVlc3RNc2c+PCFbQ0RBVEFbPD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgi

Pz4KPHJlcXVlc3QgeG1sbnM9Imh0dHA6Ly9hcGktdjEuZ2VuLm1tLnZvZGFmb25lLmNvbS9tbWlu

dGVyZmFjZS9yZXF1ZXN0Ij4KICAgIDxUcmFuc2FjdGlvbj4KICAgICAgICA8Q29tbWFuZElEPkNv

bW1hbmRJRDA8L0NvbW1hbmRJRD4KICAgICAgICA8TGFuZ3VhZ2VDb2RlPkxhbmd1YWdlQ29kZTA8

L0xhbmd1YWdlQ29kZT4KICAgICAgICA8T3JpZ2luYXRvckNvbnZlcnNhdGlvbklEPk9yaWdpbmF0

b3JDb252ZXJzYXRpb25JRDA8L09yaWdpbmF0b3JDb252ZXJzYXRpb25JRD4KICAgICAgICA8Q29u

dmVyc2F0aW9uSUQ+Q29udmVyc2F0aW9uSUQwPC9Db252ZXJzYXRpb25JRD4KICAgICAgICA8UmVt

YXJrPlJlbWFyazA8L1JlbWFyaz4KICAgICAgICA8RW5jcnlwdGVkUGFyYW1ldGVycz5FbmNyeXB0

ZWRQYXJhbWV0ZXJzMDwvRW5jcnlwdGVkUGFyYW1ldGVycz4KICAgICAgICA8UGFyYW1ldGVycz4K

ICAgICAgICAgICAgPFBhcmFtZXRlcj4KICAgICAgICAgICAgICAgIDxLZXk+S2V5MDwvS2V5Pgog

ICAgICAgICAgICAgICAgPFZhbHVlPlZhbHVlMDwvVmFsdWU+CiAgICAgICAgICAgIDwvUGFyYW1l

dGVyPgogICAgICAgICAgICA8UGFyYW1ldGVyPgogICAgICAgICAgICAgICAgPEtleT5LZXkxPC9L

ZXk+CiAgICAgICAgICAgICAgICA8VmFsdWU+VmFsdWUxPC9WYWx1ZT4KICAgICAgICAgICAgPC9Q

YXJhbWV0ZXI+CiAgICAgICAgPC9QYXJhbWV0ZXJzPgogICAgICAgIDxSZWZlcmVuY2VEYXRhPgog

ICAgICAgICAgICA8UmVmZXJlbmNlSXRlbT4KICAgICAgICAgICAgICAgIDxLZXk+UXVldWVUaW1l

b3V0VVJMPC9LZXk+CiAgICAgICAgICAgICAgICA8VmFsdWU+VmFsdWUyPC9WYWx1ZT4KICAgICAg

ICAgICAgPC9SZWZlcmVuY2VJdGVtPgogICAgICAgICAgICA8UmVmZXJlbmNlSXRlbT4KICAgICAg

ICAgICAgICAgIDxLZXk+S2V5MzwvS2V5PgogICAgICAgICAgICAgICAgPFZhbHVlPlZhbHVlMzwv

VmFsdWU+CiAgICAgICAgICAgIDwvUmVmZXJlbmNlSXRlbT4KICAgICAgICA8L1JlZmVyZW5jZURh

dGE+CiAgICAgICAgPFRpbWVzdGFtcD4KICAgICAgICA8L1RpbWVzdGFtcD4KICAgIDwvVHJhbnNh

Y3Rpb24+CiAgICA8SWRlbnRpdHk+CiAgICAgICAgPENhbGxlcj4KICAgICAgICAgICAgPENhbGxl

clR5cGU+MDwvQ2FsbGVyVHlwZT4KICAgICAgICAgICAgPFRoaXJkUGFydHlJRD5UaGlyZFBhcnR5

SUQwPC9UaGlyZFBhcnR5SUQ+CiAgICAgICAgICAgIDxQYXNzd29yZD5QYXNzd29yZDA8L1Bhc3N3

b3JkPgogICAgICAgICAgICA8Q2hlY2tTdW0+Q2hlY2tTdW0wPC9DaGVja1N1bT4KICAgICAgICAg

ICAgPFJlc3VsdFVSTD5SZXN1bHRVUkwwPC9SZXN1bHRVUkw+CiAgICAgICAgPC9DYWxsZXI+CiAg

ICAgICAgPEluaXRpYXRvcj4KICAgICAgICAgICAgPElkZW50aWZpZXJUeXBlPjE8L0lkZW50aWZp

ZXJUeXBlPgogICAgICAgICAgICA8SWRlbnRpZmllcj5JZGVudGlmaWVyMDwvSWRlbnRpZmllcj4K

ICAgICAgICAgICAgPFNlY3VyaXR5Q3JlZGVudGlhbD5TZWN1cml0eUNyZWRlbnRpYWwwPC9TZWN1

cml0eUNyZWRlbnRpYWw+CiAgICAgICAgICAgIDxTaG9ydENvZGU+CiAgICAgICAgICAgIDwvU2hv

cnRDb2RlPgogICAgICAgIDwvSW5pdGlhdG9yPgogICAgICAgIDxQcmltYXJ0eVBhcnR5PgogICAg

ICAgICAgICA8SWRlbnRpZmllclR5cGU+MTwvSWRlbnRpZmllclR5cGU+CiAgICAgICAgICAgIDxJ

ZGVudGlmaWVyPklkZW50aWZpZXIxPC9JZGVudGlmaWVyPgogICAgICAgICAgICA8U2hvcnRDb2Rl

PlNob3J0Q29kZTA8L1Nob3J0Q29kZT4KICAgICAgICA8L1ByaW1hcnR5UGFydHk+CiAgICAgICAg

PFJlY2VpdmVyUGFydHk+CiAgICAgICAgICAgIDxJZGVudGlmaWVyVHlwZT4xPC9JZGVudGlmaWVy

VHlwZT4KICAgICAgICAgICAgPElkZW50aWZpZXI+SWRlbnRpZmllcjI8L0lkZW50aWZpZXI+CiAg

ICAgICAgICAgIDxTaG9ydENvZGU+U2hvcnRDb2RlMTwvU2hvcnRDb2RlPgogICAgICAgIDwvUmVj

ZWl2ZXJQYXJ0eT4KICAgICAgICA8QWNjZXNzRGV2aWNlPgogICAgICAgICAgICA8SWRlbnRpZmll

clR5cGU+MTwvSWRlbnRpZmllclR5cGU+CiAgICAgICAgICAgIDxJZGVudGlmaWVyPklkZW50aWZp

ZXIzPC9JZGVudGlmaWVyPgogICAgICAgIDwvQWNjZXNzRGV2aWNlPgogICAgPC9JZGVudGl0eT4K

ICAgIDxLZXlPd25lcj4wPC9LZXlPd25lcj4KPC9yZXF1ZXN0Pl1dPjwvcmVxOlJlcXVlc3RNc2c+

CiAgIDwvc29hcGVudjpCb2R5Pgo8L3NvYXBlbnY6RW52ZWxvcGU+Cg==</loc:originRequest>

</loc:notifyQueueTimeout>

</soapenv:Body>

</soapenv:Envelope>

### The 3rd party return response to the Broker

Example1: Success response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org/schema/timeoutnotification/data/v1\_0/local" xmlns:res="http://api-v1.gen.mm.vodafone.com/mminterface/result">

<soapenv:Header/>

<soapenv:Body>

<loc:notifyQueueTimeoutResponse>

<loc:result>

<res:ResultCode>00000000</res:ResultCode>

<res:ResultDesc> success</res:ResultDesc>

</loc:result>

</loc:notifyQueueTimeoutResponse>

</soapenv:Body>

</soapenv:Envelope>

Example2: Error response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org/schema/timeoutnotification/data/v1\_0/local" xmlns:res="http://api-v1.gen.mm.vodafone.com/mminterface/result">

<soapenv:Header/>

<soapenv:Body>

<loc:notifyQueueTimeoutResponse>

<loc:result>

<res:ResultCode>000000001</res:ResultCode>

<res:ResultDesc> failed </res:ResultDesc>

</loc:result>

</loc:notifyQueueTimeoutResponse>

</soapenv:Body>

</soapenv:Envelope>

## queryTransaction

### The 3rd party sends query request to the Broker

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:v2="http://www.huawei.com.cn/schema/common/v2\_1" xmlns:loc="http://www.csapi.org/schema/transaction/data/v1\_0/local" xmlns:res="http://api-v1.gen.mm.vodafone.com/mminterface/result">

<soapenv:Header>

<v2:RequestSOAPHeader>

<v2:spId>151515</v2:spId>

<v2:spPassword>ODExNERCMDlCNjVDRkYxQUFCNzE5MkE1OEQyMjJDMzk1RUFCNzgwMjk2ODE4Rjk1OTE2MEFGNDU1QkRCMDkyMg==</v2:spPassword>

<v2:serviceId>151515000</v2:serviceId>

<v2:timeStamp>20140730093620</v2:timeStamp>

</v2:RequestSOAPHeader>

</soapenv:Header>

<soapenv:Body>

<loc:queryTransaction>

<loc:originatorConversationID>R9I1-0000-4225-g64153</loc:originatorConversationID>

<loc:extensionInfo>

<loc:item>

<res:Key>queryDate</res:Key>

<res:Value>20140930102022</res:Value>

</loc:item>

</loc:extensionInfo>

</loc:queryTransaction>

</soapenv:Body>

### The Broker return response to the 3rd party

Example1: Success response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>

<ns1:queryTransactionResponse xmlns:ns1="http://www.csapi.org/schema/transaction/data/v1\_0/local">

<ns1:result>

<ns2:ResponseCode xmlns:ns2="http://api-v1.gen.mm.vodafone.com/mminterface/response">00000000</ns2:ResponseCode>

<ns3:ResponseDesc xmlns:ns3="http://api-v1.gen.mm.vodafone.com/mminterface/response">Success</ns3:ResponseDesc>

</ns1:result>

<ns1:submitApiRequestList>

<ns1:submitApiRequest></ns1:submitApiRequest>

<ns1:submitApiRequest></ns1:submitApiRequest>

</ns1:submitApiRequestList>

<ns1:submitApiResponseList>

<ns1:submitApiResponse></ns1:submitApiResponse>

<ns1:submitApiResponse></ns1:submitApiResponse>

</ns1:submitApiResponseList>

<ns1:submitApiResultList>

<ns1:submitApiResult></ns1:submitApiResult>

</ns1:submitApiResultList>

<ns1:queueTimeOutList/>

</ns1:queryTransactionResponse>

</soapenv:Body>

</soapenv:Envelope>Example2: Error response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org/schema/transaction/data/v1\_0/local" xmlns:res="http://api-v1.gen.mm.vodafone.com/mminterface/response" xmlns:res1="http://api-v1.gen.mm.vodafone.com/mminterface/result">

<soapenv:Header/>

<soapenv:Body>

<loc:queryTransactionResponse>

<loc:result>

<ResponseCode>100000007</ResponseCode>

<ResponseDesc>Authentication failed</ResponseDesc>

</loc:result>

</loc:queryTransactionResponse>

</soapenv:Body>

</soapenv:Envelope>

## changePassword

### Request example:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org/schema/management/data/v1\_0/local" xmlns:res="http://api-v1.gen.mm.vodafone.com/mminterface/result">

<soapenv:Header/>

<soapenv:Body>

<loc:changePassword>

<loc:spId>3500001</loc:spId>

<loc:spPassword>c5216e519a071d601bedd150f3fcd026</loc:spPassword>

<loc:timeStamp>20120101010101</loc:timeStamp>

<loc:newPassword>wi2a7BAH0QPd2LRdmcgC9w==</loc:newPassword>

</loc:changePassword>

</soapenv:Body>

</soapenv:Envelope>

### Response example:

* Example1: Success response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org/schema/transaction/data/v1\_0/local" xmlns:res="http://api-v1.gen.mm.vodafone.com/mminterface/response" xmlns:res1="http://api-v1.gen.mm.vodafone.com/mminterface/result">

<soapenv:Header/>

<soapenv:Body>

<loc:changePassword Response>

<loc:result>

<res:ResponseCode>00000000</res:ResponseCode>

<res:ResponseDesc>Success</res:ResponseDesc>

</loc:result>

<loc:changePasswordResponse>

</soapenv:Body>

</soapenv:Envelope>

* Example2: Error response caused by authentication failed.

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:req="http://api-v1.gen.mm.vodafone.com/mminterface/request">

<soapenv:Header/>

<soapenv:Body>

<req:ResponseMsg><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<response xmlns="http://api-v1.gen.mm.vodafone.com/mminterface/response">

<ResponseCode>100000007</ResponseCode>

<ResponseDesc>Authentication failed</ResponseDesc>

</response>]]></req:ResponseMsg>

</soapenv:Body>

</soapenv:Envelope>