

|  |
| --- |
| Card Management System  Implementation Guide |
| Version 1.0 |
| **Issued: July 2017** |

Table of Contents

[1. Revision History 2](#_Toc487527164)

[2. Definitions, Acronyms, and Abbreviations 2](#_Toc487527165)

[3. Introduction 3](#_Toc487527166)

[4. Data Element and Attribute 4](#_Toc487527167)

[5. SOAP 5](#_Toc487527168)

[5.1 What’s Web Service? 5](#_Toc487527169)

[5.2 What’s SOAP? 5](#_Toc487527170)

[5.3 SOAP - Message Structure 6](#_Toc487527171)

[5.3.1 SOAP - Envelope Element 7](#_Toc487527172)

[5.3.2 SOAP - Header Element 9](#_Toc487527173)

[5.3.3 SOAP - Body Element 10](#_Toc487527174)

[5.3.4 SOAP - Fault Element 11](#_Toc487527175)

[5.4 SOAP-Encoding 14](#_Toc487527176)

[5.5 SOAP – Transport 14](#_Toc487527177)

[6. CMS Services 15](#_Toc487527178)

[6.1 CMS Services 15](#_Toc487527179)

[6.1.1 Online CMS 15](#_Toc487527180)

[7. Online CMS 17](#_Toc487527181)

[7.1 Message Flows 18](#_Toc487527189)

[7.1.1 Successful Inquire Customer Cards 18](#_Toc487527190)

[7.1.2 Successful Inquire Credit Card 18](#_Toc487527191)

[7.1.3 Successful Restrict Credit Card 19](#_Toc487527192)

[7.1.4 Successful Restrict Debit Card 19](#_Toc487527193)

[7.1.5 Successful Inquire Credit card Last Statement 20](#_Toc487527194)

[7.1.6 Successful Transfer to Credit card 20](#_Toc487527195)

[7.1.7 Successful Inquire Credit Card Transactions by Interval 21](#_Toc487527196)

[7.1.8 Successful Inquire Credit Card Transactions by Count 21](#_Toc487527197)

[7.2 Operations 22](#_Toc487527198)

[7.2.1 CMS Operations 22](#_Toc487527199)

[8. Security 58](#_Toc487527200)

[8.1 Authentication 58](#_Toc487527210)

[8.2 Digital Signature 58](#_Toc487527211)

[9. Appendixes 60](#_Toc487527212)

[9.1 Appendix A 60](#_Toc487527213)

[9.2 Appendix B 60](#_Toc487527214)

# Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Created By** | **Linguistic and Formatting Review** | **Technical Review** | **Description** |
| 1.0 | 07/02/2017 | Joseph Gamal |  |  | Draft Version |

# Definitions, Acronyms, and Abbreviations

| **#** | **Definition** | **Description** |
| --- | --- | --- |
| 1 | CMS | Card Management System |
| 2 | SOAP | Simple Object Access Protocol |
| 3 | WSDL | [Web Services Description Language](https://en.wikipedia.org/wiki/Web_Services_Description_Language) |
| 4 | HTTP | Hypertext Transfer Protocol |
| 5 | PKI | Public Key Infrastructure |
| 6 | RSA | One of the First Practical Public-Key Cryptosystems |

# Introduction

EBC Card Management System is the bank's way to manage its cards.

This document serves as the technical reference for the specification of Card Management system. The document defines flows and exchanges that are needed for banks to manage cards and execute transactions. The document defines use cases and their associated message exchanges.

This document should be the reference for the technical team that will work on the development required for the integration with the Card Management System. The following sections will describe in detail the technology, protocols and different use cases of the CMS messages implementation.

# Data Element and Attribute

All request and response messages are composed of one or more standard high level data elements. Depending upon the type of service interface, some data elements may be required or may be optional. The mandatory or optional requirement is indicated by the allowed number of occurrence:

0 - 1 = Optional or conditional (refer to field comment), and may exist once only

1 - 1 = mandatory, and must exist once only

1 - N = mandatory, and can exist once or more (up to n times)

\* Please refer to fields comments as some fields are optional but must be set in some circumstances

All data elements types are standard XML datatypes and described in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Data Type** | **Description** |  |
|  | Int | Data type is used to represent positive integer values. |  |
|  | Decimal | Data type is used to represent numeric values. It supports positive decimal numbers up to 16 digits and 2 fractions. |  |
|  | date | Data type is used to represent date in YYYY-MM-DD format. |  |
|  | String | Data type can take characters, line feeds, carriage returns, and tab characters. |  |
|  | Date Time ( UTC ) | Data type is used to represent date and time in YYYY-MM-DDThh:mm:ssZ format. |  |
|  | GUID | Global Unique Identifier , is an implementation of UUID ( Universally Unique Identifier ) , 36 characters long following the following pattern :  [a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12} |  |

# SOAP

SOAP Version 1.2 is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics.

**List of standards:**

* ISO/IEC DIS 40210,Information technology – W3C SOAP Version 1.2 Part 1: Messaging Framework
* ISO/IEC DIS 40220, Information technology – W3C SOAP Version 1.2 Part 2: Adjuncts
* ISO/IEC DIS 40230,Information technology – W3C SOAP Message Transmission Optimization Mechanism
* ISO/IEC DIS 40240, Information technology – W3C Web Services Addressing 1.0 – Core
* ISO/IEC DIS 40250,Information technology – W3C Web Services Addressing 1.0 – SOAP Binding
* ISO/IEC DIS 40260,Information technology – W3C Web Services Addressing 1.0 – Metadata
* ISO/IEC DIS 40270,Information technology – W3C Web Services Policy 1.5 – Framework
* ISO/IEC DIS 40280,Information technology – W3C Web Services Policy 1.5 – Attachment

## What’s Web Service?

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

## What’s SOAP?

SOAP is an XML-based messaging protocol. It defines a set of rules for structuring messages that can be used for simple one-way messaging but is particularly useful for performing RPC-style (Remote Procedure Call) request-response dialogues. It is not tied to any particular transport protocol though HTTP is popular. Nor is it tied to any particular operating system or programming language so theoretically the clients and servers in these dialogues can be running on any platform and written in any language as long as they can formulate and understand SOAP messages. As such it is an important building block for developing distributed applications that exploit functionality published as services over an intranet or the internet.

**Points to note**

Below mentioned are some important point which the user should take note of. These points briefly describe the nature of SOAP:

* SOAP is a communication protocol designed to communicate via Internet.
* SOAP can extend HTTP for XML messaging.
* SOAP provides data transport for Web services.
* SOAP can exchange complete documents or call a remote procedure.
* SOAP can be used for broadcasting a message.
* SOAP is platform- and language-independent
* SOAP is the XML way of defining what information is sent and how.
* SOAP enables client applications to easily connect to remote services and invoke remote methods.

## SOAP - Message Structure

SOAP message is an ordinary XML document containing the following elements:

* **Envelope** − Defines the start and the end of the message. It is a mandatory element.
* **Header** − Contains any optional attributes of the message used in processing the message, either at an intermediary point or at the ultimate end-point. It is an optional element.
* **Body** − Contains the XML data comprising the message being sent. It is a mandatory element.
* **Fault** − an optional Fault element that provides information about errors that occur while processing the message.

All these elements are declared in the default namespace for the SOAP envelope:

[https://www.w3.org/2001/12/soap-envelope](https://www.w3.org/2001/12/soap-envelope )

And the default namespace for SOAP encoding and data types is:

<http://www.w3.org/2001/12/soap-encoding>

The following block depicts the general structure of a SOAP message

|  |
| --- |
| <?xml version="1.0"?>  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2001/12/soap-envelope" SOAP-ENV:encodingStyle="http://www.w3.org/2001/12/soap-encoding">  <SOAP-ENV:Header>  ...  ...  </SOAP-ENV:Header>  <SOAP-ENV:Body>  ...  ...  <SOAP-ENV:Fault>  ...  ...  </SOAP-ENV:Fault>  ...  </SOAP-ENV:Body>  </SOAP\_ENV:Envelope> |

Figure 1 Structure of a SOAP message

### SOAP - Envelope Element

The SOAP envelope indicates the start and the end of the message so that the receiver knows when an entire message has been received. The SOAP envelope solves the problem of knowing when you are done receiving a message and are ready to process it.

The SOAP envelope is therefore basically a packaging mechanism.

**Points to note**

Below mentioned are some important points about SOAP envelope element:

* Every SOAP message has a root Envelope element.
* Envelope is a mandatory part of SOAP message.
* Every Envelope element must contain exactly one Body element.
* If an Envelope contains a Header element, it must contain no more than one, and it must appear as the first child of the Envelope, before the Body.
* The envelope changes when SOAP versions change.
* The SOAP envelope is specified using the *ENV* namespace prefix and the Envelope element.
* The optional SOAP encoding is also specified using a namespace name and the optional *encodingStyle* element, which could also point to an encoding style other than the SOAP one.
* A v1.1-compliant SOAP processor generates a fault upon receiving a message containing the v1.2 envelope namespace.
* A v1.2-compliant SOAP processor generates a *VersionMismatch* fault if it receives a message that does not include the v1.2 envelope namespace.

Given below is an example of v1.2-compliant SOAP message:

|  |
| --- |
| <?xml version="1.0"?>  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2001/12/soap-envelope" SOAP-ENV:encodingStyle=" http://www.w3.org/2001/12/soap-encoding">  ...  Message information goes here  ...  </SOAP-ENV:Envelope> |

Figure 2 v1.2-compliant SOAP message

### SOAP - Header Element

The optional Header element offers a flexible framework for specifying additional application-level requirements.

**Points to note**

Below mentioned are few important points about SOAP header element to take note of:

* It is an optional part of a SOAP message.
* Header elements can occur multiple times.
* Headers are intended to add new features and functionality.
* The SOAP header contains header entries defined in a namespace.
* The header is encoded as the first immediate child element of the SOAP envelope.
* When multiple headers are defined, all immediate child elements of the SOAP header are interpreted as SOAP header blocks.

#### Soap Header Attribute

Actor attribute

The SOAP protocol defines a message path as a list of SOAP service nodes. Each of these intermediate nodes can perform some processing and then forward the message to the next node in the chain. By setting the Actor attribute, the client can specify the recipient of the SOAP header.

MustUnderstand attribute

It indicates whether a Header element is optional or mandatory. If set to true, the recipient must understand and process the Header attribute according to its defined semantics, or return a fault.

|  |
| --- |
| <?xml version="1.0"?>  <SOAP-ENV:Envelope xmlns:SOAP-ENV=" http://www.w3.org/2001/12/soap-envelope" SOAP-ENV:encodingStyle=" http://www.w3.org/2001/12/soap-encoding">  <SOAP-ENV:Header>  <t:Transaction xmlns:t="http://www.tutorialspoint.com/transaction/" SOAP-ENV:mustUnderstand="true">5</t:Transaction>  </SOAP-ENV:Header>  ...  ...  </SOAP-ENV:Envelope> |

Figure 3 Soap Message Header

### SOAP - Body Element

The SOAP body is a mandatory element that contains the application-defined XML data being exchanged in the SOAP message. The body must be contained within the envelope and must follow any headers that might be defined for the message.

The body is defined as a child element of the envelope, and the semantics for the body are defined in the associated SOAP schema.

The body contains mandatory information intended for the ultimate receiver of the message. For example:

|  |
| --- |
| <?xml version="1.0"?>  <SOAP-ENV:Envelope>  ........  <SOAP-ENV:Body>  <m:LoadCustomerName xmlns:m="http://www.xyz.com/Customer">  <m:Item>5</m:Item>  </m:LoadCustomerName>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

Figure 4 Soap Message Body (Request)

The example above requests a quotation of computer sets. Note that the m:LoadCustomerName and the Item elements above are application-specific elements. They are not a part of the SOAP standard.

Here is the response to the above query:

|  |
| --- |
| <?xml version="1.0"?>  <SOAP-ENV:Envelope>  ........  <SOAP-ENV:Body>  <m:LoadCustomerNameResponse xmlns:m="http://www.xyz.com/Customer">  <m:Quotation>Customer</m:Quotation>  </m:LoadCustomerNameResponse>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

Figure 5 Soap Message Body (Response)

### SOAP - Fault Element

If an error occurs during processing, the response to a SOAP message is a SOAP fault element in the body of the message, and the fault is returned to the sender of the SOAP message.

The SOAP fault mechanism returns specific information about the error, including a predefined code, a Description, and the address of the SOAP processor that generated the fault.

**Points to note**

Below mentioned are few important points about SOAP fault element to take note of:

* A SOAP message can carry only one fault block.
* Fault is an optional part of a SOAP message.
* For HTTP binding, a successful response is linked to the 200 to 299 range of status codes.
* SOAP Fault is linked to the 500 to 599 range of status codes.

#### Sub-elements of Fault

The SOAP Fault has the following sub elements:

|  |  |
| --- | --- |
| **Sub-element** | **Description** |
| <faultCode> | It is a text code used to indicate a class of errors. See the next Table for a listing of predefined fault codes. |
| <faultString> | It is a text message explaining the error. |
| <faultActor> | It is a text string indicating who caused the fault. It is useful if the SOAP message travels through several nodes in the SOAP message path, and the client needs to know which node caused the error. A node that does not act as the ultimate destination must include a *faultActor* element. |
| <detail> | It is an element used to carry application-specific error messages. The detail element can contain child elements called detail entries. |

Table 1 Fault Sub Elements

#### SOAP Fault Codes

The faultCode values defined below must be used in the *faultcode* element while describing faults.

|  |  |
| --- | --- |
| **Error** | **Description** |
| SOAP-ENV:VersionMismatch | Found an invalid namespace for the SOAP Envelope element. |
| SOAP-ENV:MustUnderstand | An immediate child element of the Header element, with the mustUnderstand attribute set to "1", was not understood. |
| SOAP-ENV:Client | The message was incorrectly formed or contained incorrect information. |
| SOAP-ENV:Server | There was a problem with the server, so the message could not proceed. |

Table 2 Fault Codes

#### SOAP Fault Example

The following code is a sample Fault. The client has requested a method named *ValidateCreditCard*, but the service does not support such a method. This represents a client request error, and the server returns the following SOAP response:

|  |
| --- |
| <?xml version='1.0' encoding='UTF-8'?>  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:xsd="http://www.w3.org/1999/XMLSchema">  <SOAP-ENV:Body>  <SOAP-ENV:Fault>  <faultcode xsi:type="xsd:string">SOAP-ENV:Client</faultcode>  <faultstring xsi:type="xsd:string">  Failed to locate method (ValidateCreditCard) in class (TestCreditCard) at /class/SOAP/TestCreditCard.cs line 1555.  </faultstring>  </SOAP-ENV:Fault>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

Figure 6 Soap Fault Example

## SOAP-Encoding

SOAP includes a built-in set of rules for encoding data types.

It enables the SOAP message to indicate specific data types, such as integers, floats, doubles, or arrays.

* SOAP data types are divided into two broad categories − scalar types and compound types.
* Scalar types contain exactly one value such as a last name, price, or product Description.
* Compound types contain multiple values such as a purchase order or a list of stock quotes.
* Compound types are further subdivided into arrays and structs.
* The encoding style for a SOAP message is set via the *SOAP-ENV: encodingStyle* attribute.
* To use SOAP 1.2 encoding, use the value <http://www.w3.org/2001/12/soap-encoding>.
* Latest SOAP specification adopts all the built-in types defined by XML Schema. Still, SOAP maintains its own convention for defining constructs not standardized by XML Schema, such as arrays and references.

## SOAP – Transport

SOAP specification includes details on HTTP only. HTTP remains the most popular SOAP transport protocol.Quite logically, SOAP requests are sent via an HTTP request and SOAP responses are returned within the content of the HTTP response. While SOAP requests can be sent via an HTTP GET, the specification includes details on HTTP POST only.

Additionally, both HTTP requests and responses are required to set their content type to text/xml.

# CMS Services

The communication between banks and Card Management system is based on SOAP web services technology. Banks must implement invoking services to send all financial and non-financial messages to Card Management system services. The below sections list all services operations.

**CMS services** list all operations provided by Card Management System to be invoked by banks.

## CMS Services

### Online CMS

#### Inquire Customer Cards

|  |  |
| --- | --- |
| WSDL | OnlineCMS |
| Service | OnlineCMSService |
| Operation | InquireCustomerCards |
| Message | ClientId , InquireCustomerCardsRequest |
| InquireCustomerCardsResponse |

#### Inquire Credit Card

|  |  |
| --- | --- |
| WSDL | OnlineCMS |
| Service | OnlineCMSService |
| Operation | InquireCreditCard |
| Message | ClientId , InquireCreditCardRequest |
| InquireCreditCardResponse |

#### Restrict Credit Card

|  |  |
| --- | --- |
| WSDL | OnlineCMS |
| Service | OnlineCMSService |
| Operation | RestrictCreditCard |
| Message | ClientId , RestrictCreditCardRequest |
| RestrictCreditCardResponse |

#### Restrict Debit Card

|  |  |
| --- | --- |
| WSDL | OnlineCMS |
| Service | OnlineCMSService |
| Operation | RestrictDebitCard |
| Message | ClientId , RestrictDebitCardRequest |
| RestrictDebitCardResponse |

#### Inquire Credit Card Transactions by Interval

|  |  |
| --- | --- |
| WSDL | OnlineCMS |
| Service | OnlineCMSService |
| Operation | InquireCreditCardTransactionsByInterval |
| Message | ClientId , InquireCreditCardTransactionsByIntervalRequest |
| InquireCreditCardTransactionsByIntervalResponse |

#### Inquire Credit Card Transactions by Count

|  |  |
| --- | --- |
| WSDL | OnlineCMS |
| Service | OnlineCMSService |
| Operation | InquireCreditCardTransactionsByCount |
| Message | ClientId , InquireCreditCardTransactionsByCountRequest |
| InquireCreditCardTransactionsByCountResponse |

#### Inquire Credit Card Last Statement

|  |  |
| --- | --- |
| WSDL | OnlineCMS |
| Service | OnlineCMSService |
| Operation | InquireCreditCardLastStatement |
| Message | ClientId , InquireCreditCardLastStatementRequest |
| InquireCreditCardLastStatementResponse |

#### Transfer to Credit Card

|  |  |
| --- | --- |
| WSDL | OnlineCMS |
| Service | OnlineCMSService |
| Operation | TransferToCreditCard |
| Message | ClientId , TransferToCreditCardRequest |
| TransferToCreditCardResponse |

# Online CMS

EBC provides Card Management service to avail the following operations in online modes

* Inquire Customer Cards.
* Inquire Credit Card.
* Inquire Credit Card Transactions By Interval.
* Inquire Credit Card Transactions By Count.
* Inquire Credit Card Last Statement.
* Restrict Credit Card.
* Restrict Debit Card.
* Transfer to Credit Card.



## Message Flows

### Successful Inquire Customer Cards



1. Bank A initiates "Inquire Customer Cards" request to online CMS.
2. Online CMS validates incoming request and processes it then sends response and list of cards details.

### Successful Inquire Credit Card



1. Bank A initiates "Inquire Credit Card" request to online CMS.
2. Online CMS validates incoming request and processes it then sends response.

### Successful Restrict Credit Card



1. Bank A initiates "Restrict Credit Card" request to online CMS.
2. Online CMS validates incoming request and processes it then sends response.

### Successful Restrict Debit Card



1. Bank A initiates "Restrict Debit Card" request to online CMS.
2. Online CMS validates incoming request and processes it then sends response.

### Successful Inquire Credit card Last Statement



1. Bank A initiates "Inquire Credit Card Last Statement" request to online CMS.
2. Online CMS validates incoming request and processes it then sends response.

### Successful Transfer to Credit card



1. Bank A initiates "Transfer to Credit Card" request to online CMS.
2. Online CMS validates incoming request and processes it then sends response.

### Successful Inquire Credit Card Transactions by Interval



1. Bank A initiates "Inquire Credit Card Transactions by Interval" request to online CMS.
2. Online CMS validates incoming request and processes it then sends response.

### Successful Inquire Credit Card Transactions by Count



1. Bank A initiates “Inquire Credit Card Transactions by Count" request to online CMS.
2. Online CMS validates incoming request and processes it then sends response.

## Operations

### CMS Operations

#### Online CMS Operations

##### Inquire Customer Cards Operation

Avails inquire customer cards in CMS

###### InquireCustomerCardsRequest

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Created by sender * Must be unique among the whole CMS requests * Formatted as GUID |
| Channel |  | string | 6 - 8 | 1 – 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| CustomerId |  | String | 1 – 15 | 1 – 1 | * Bank set Unique customer Id for every customer in Debit or credit Management system * Must be unique among the whole customers |

**Sample:**

*<Message>*

<*Signature*> [signature](#_SIGNING) here <*/*Signature>

<CMS.Inquire.Customer.Cards.Request>

<Data>

<Header>

<Version>1.00</Version>

<ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId >

<RequestId>00000000-0000-0000-0000-000000000000</RequestId>

<Channel>mobile</Channel>

</Header>

<Body>

<CustomerId>932432421</CustomerId>

</Body>

</Data>

</CMS.Inquire.Customer.Cards.Request>

*</Message*>

**InquireCustomerCardsResponse**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Same as RequestId sent in request |
| Channel |  | string | 6 - 8 | 1 – 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| ResponseCode |  | String | 5 – 5 | 1 – 1 | * 5 numbers left padded by zeros |
| ResponseDescription |  | String | 1 – 1024 | 1 – 1 |  |
| **Cards** | |  |  | 0 – 1 |  |
| **Card** | |  |  | 1 – \* |  |
| CardNumber |  | String | 16 – 19 | 1 – 1 | * represents card number |
| ProductType |  | String | 5 – 6 | 1 – 1 | * (Debit, Credit) Only allowed values |
| CardProfile |  | String | 7 – 15 | 1 – 1 | * (Primary, Supplementary) Only allowed values |
| CardStatus |  | String | 1 – 3 | 1 – 1 | * Appendix A |

**Sample 1 (Success with Data Found):**

|  |
| --- |
| <message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Inquire.Customer.Cards.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>Success</ResponseDescription>  <Cards>  <Card>  <CardNumber>5118630102038108</CardNumber>  <ProductType>Debit</ProductType>  <CardProfile>Primary</CardProfile>  <CardStatus>1</CardStatus>  </Card>  <Card>  <CardNumber>5118630102038108</CardNumber>  <ProductType>Credit</ProductType>  <CardProfile>Primary</CardProfile>  <CardStatus>1</CardStatus>  </Card>  <Card>  <CardNumber>5118630102038108</CardNumber>  <ProductType>Debit</ProductType>  <CardProfile>Primary</CardProfile>  <CardStatus>9</CardStatus>  </Card>  <Card>  <CardNumber>5118630102038108</CardNumber>  <ProductType>Credit</ProductType>  <CardProfile>Primary</CardProfile>  <CardStatus>9</CardStatus>  </Card>  <Card>  <CardNumber>5118630102038108</CardNumber>  <ProductType>Debit</ProductType>  <CardProfile>Supplementary</CardProfile>  <CardStatus>1</CardStatus>  </Card>  <Card>  <CardNumber>5118630102038108</CardNumber>  <ProductType>Credit</ProductType>  <CardProfile>Supplementary</CardProfile>  <CardStatus>1</CardStatus>  </Card>  <Card>  <CardNumber>5118630102038108</CardNumber>  <ProductType>Debit</ProductType>  <CardProfile>Supplementary</CardProfile>  <CardStatus>9</CardStatus>  </Card>  <Card>  <CardNumber>5118630102038108</CardNumber>  <ProductType>Credit</ProductType>  <CardProfile>Supplementary</CardProfile>  <CardStatus>9</CardStatus>  </Card>  </Cards>  </Body>  </Data>  </CMS.Inquire.Customer.Cards.Response>  </message> |

**Sample 2 (Success with No Data Found):**

|  |
| --- |
| <message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Inquire.Customer.Cards.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>No Data Found</ResponseDescription>  </Body>  </Data>  </CMS.Inquire.Customer.Cards.Response>  </message> |

**Sample 3 (Fail):**

|  |
| --- |
| <message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Inquire.Customer.Cards.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>31001</ResponseCode>  <ResponseDescription>customer not found</ResponseDescription>  </Body>  </Data>  </CMS.Inquire.Customer.Cards.Response>  </message> |

##### Inquire Credit Card Operation

Avails inquire credit card in CMS

| **InquireCreditCardRequest** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Created by sender * Must be unique among the whole CMS requests * Formatted as GUID |
| Channel |  | string | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| CardNumber |  | String | 16 - 19 | 1 – 1 | * Number of credit card   Ex. 5118630102038108 |

**Sample:**

|  |
| --- |
| <Message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Inquire.Credit.Card.Request>  <Data>  <Header>  <Version>1.00</Version>  <ClientID>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientID>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <CardNumber>5118630102038108</CardNumber>  </Body>  </Data>  </CMS.Inquire.Credit.Card.Request>  </Message> |

###### InquireCreditCardresponse

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | **1 – 1** |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Same as RequestId sent in request |
| Channel |  | String | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| ResponseCode |  | String | 5 - 5 | 1 - 1 | * 5 numbers left padded by zeros |
| ResponseDescription |  | String | 1-1024 | 1 - 1 |  |
| CardStatus |  | String | 3 - 3 | 1 - 1 | * Appendix A |
| LedgerBalance |  | Decimal | 16 , 2 | 1 - 1 | * Ledger balance is your current bank balance. |
| AvailableBalance |  | Decimal | 16 , 2 | 1 - 1 | * available balance includes credits or debits from transactions that have not yet posted to your account |
| AccountCurrency |  | String | 3 - 3 | 1 - 1 | * (818,840) allowed values |
| CardholderName |  | string | 1 - 30 | 1 - 1 | * Name of the card Holder |
| CardProfile |  | String | 1 - 50 | 1 - 1 | * Name of the card profile   Ex. Gold, Classic... etc. |
| CreditLimit |  | Decimal | 16 , 2 | 1 - 1 | * Limit of the credit card |

**Sample 1 (Successful):**

|  |
| --- |
| <Message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Inquire.Credit.Cards.Response>  <Data>  <Header>  <Version>1.00</Version>  </Header>  <Body>  <CardStatus>0</CardStatus>  <LedgerBalance>600</LedgerBalance>  <Availablebalance>600</Availablebalance>  <AccountCurrency>818</AccountCurrency>  <CardholderName>Joseph Gamal</CardholderName>  <CardProfile>Gold</CardProfile>  <CreditLimit>20000</CreditLimit>  <ResponseCode>00000</ResponseCode>  < ResponseDescription >success< ResponseDescription/>  </Body>  </Data>  </ CMS.Inquire.Credit.Cards.Response>  </Message> |

**Sample 2 (Credit card not found):**

|  |
| --- |
| <Message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Inquire.Credit.Cards.Response>  <Data>  <Header>  <Version>1.00</Version>  </Header>  <Body>  <ResponseCode>31002</ResponseCode>  <ResponseDescription>Card not found<ResponseDescription/>  </Body>  </Data>  </CMS.Inquire.Credit.Cards.Response>  </Message> |

##### Restrict Credit Card Operation

Avails Restrict Credit card in CMS

##### Restrict Credit Card Request

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClinetId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Created by sender * Must be unique among the whole CMS requests * Formatted as GUID |
| Channel |  | String | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| CardNumber |  | String | 16 - 19 | 1 - 1 | * Card number for customer   Ex. 5118630102038108 |

**Sample:**

<*Message*>

<*Signature*> [signature](#_SIGNING) here <*/*Signature>

<CMS.Restrict.Credit.Card.Request>

<Data>

<Header>

<Version>1.00</Version>

<ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>

<RequestId>00000000-0000-0000-0000-000000000000</RequestId>

<Channel>mobile</Channel>

</Header>

<Body>

<CardNumber>5118630102038108</CardNumber>

</Body>

</Data>

</CMS.Restrict.Credit.Card.Request>

</*Message*>

**RestrictCreditCardResponse**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Same as RequestId sent in request |
| Channel |  | String | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| ResponseCode |  | String | 5 – 5 | 1 – 1 | * 5 numbers left padded by zeros |
| ReasonDescription |  | String | 1 - 1024 | 1 – 1 |  |

**Sample 1 (Successful):**

|  |
| --- |
| <message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Restrict.Credit.Card.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>Success</ResponseDescription>  </Body>  </Data>  </CMS.Restrict.Credit.Card.Response>  </message> |

**Sample 2 (Failed):**

|  |
| --- |
| <message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Restrict.Credit.Card.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00001</ResponseCode>  <ResponseDescription>card not found</ResponseDescription>  </Body>  </Data>  </CMS.Restrict.Credit.Card.Response>  </message> |

##### Restrict Debit Card Operation

Avails Restrict Debit card in CMS

##### Restrict Debit Card Request

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClinetId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Created by sender * Must be unique among the whole CMS requests * Formatted as GUID |
| Channel |  | String | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| CardNumber |  | String | 16 - 19 | 1 - 1 | * Card number for customer   Ex. 5118630102038108 |

**Sample:**

<*Message*>

<*Signature*> [signature](#_SIGNING) here <*/*Signature>

<CMS.Restrict.Debit.Card.Request>

<Data>

<Header>

<Version>1.00</Version>

<ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>

<RequestId>00000000-0000-0000-0000-000000000000</RequestId>

<Channel>mobile</Channel>

</Header>

<Body>

<CardNumber>5118630102038108</CardNumber>

</Body>

</Data>

</CMS.Restrict.Debit.Card.Request>

</*Message*>

**RestrictDebitCardResponse**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Same as RequestId sent in request |
| Channel |  | String | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| ResponseCode |  | String | 5 – 5 | 1 – 1 | * 5 numbers left padded by zeros |
| ReasonDescription |  | String | 1 - 1024 | 1 – 1 |  |

**Sample 1 (Successful):**

|  |
| --- |
| <message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Restrict.Debit.Card.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>Success</ResponseDescription>  </Body>  </Data>  </CMS.Restrict.Debit.Card.Response>  </message> |

**Sample 2 (Failed):**

|  |
| --- |
| <message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Restrict.Debit.Card.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00001</ResponseCode>  <ResponseDescription>card not found</ResponseDescription>  </Body>  </Data>  </CMS.Restrict.Debit.Card.Response>  </message> |

* + - * 1. **Inquire Credit Card Last Statement Operation**

Avails Inquire credit card last statement in CMS

**InquireCreditCardLastStatementRequest**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClinetId |  | Guid | 36 - 36 | 1 - 1 | * Give n by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Created by sender * Must be unique among the whole CMS requests * Formatted as GUID |
| Channel |  | String | 6 - 8 | 1 – 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| CardNumber |  | String | 16 - 19 | 1 – 1 | * Card number for customer   Ex. 5118630102038108 |

**Sample:**

*<Message>*

<*Signature*> [signature](#_SIGNING) here <*/*Signature>

<CMS.Inquire.Credit.Card.LastStatement.Request>

<Data>

<Header>

<Version>1.00</Version>

<ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>

<RequestId>00000000-0000-0000-0000-000000000000</RequestId>

<Channel>mobile</Channel>

</Header>

<Body>

<CardNumber>05124102154789214</CardNumber>

</Body>

</Data>

</CMS.Inquire.Credit.Card.LastStatement.Request>

*</Message>*

**InquireCreditCardLastStatementResponse**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Same as RequestId sent in request |
| Channel |  | String | 6-8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| ResponseCode |  | String | 5 – 5 | 1 – 1 | * 5 numbers left padded by zeros |
| ReasonDescription |  | String | 1 - 1024 | 1 – 1 |  |
| StartDate |  | Date | 10 -10 | 0 – 1 | * The start date for the statement |
| DueDateAmount |  | Date | 10 – 10 | 0 – 1 | * The date that must pay before it total due amount |
| TotalDueAmount |  | Decimal | 16 , 2 | 0 – 1 | * the amount you have used on your credit card and liable to pay the bank |
| NewBalance |  | Decimal | 16 , 2 | 0 – 1 | * Opening Balance |
| LastBalance |  | Decimal | 16 , 2 | 0 – 1 | * Closing Balance |
| MinimumPaymentAmount |  | Decimal | 16 , 2 | 0 – 1 | * The minimum credit amount that must be pay |
| Transactions |  |  |  | 0 - 1 |  |
| Transaction |  |  |  | 1 - \* |  |
| TransactionDate |  | Date | 10 - 10 | 1 - 1 | * Date of the transaction |
| TransactionAmount |  | Decimal | 16 , 2 | 1 - 1 | * The Mount of the transaction |
| MerchantName |  | String | 1 - 25 | 1 - 1 | * The Name of The Merchant with who this transaction was made |
| Currency |  | String | 3 - 3 | 1 - 1 | * The transaction currency   Allowed values 🡺 818 , 840 |
| TransactionDescription |  | String | 1 - 25 | 1 - 1 | * The message description for the transaction |
| LedgerEffect |  | String | 5 - 6 | 1 - 1 | * The transaction effect type   Allowed Values 🡺 Credit, debit |

**Sample 1 (Success):**

|  |
| --- |
| <message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Inquire.Credit.Card.LastStatement.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>Success</ResponseDescription>  <StartDate>2017-06-01</StartDate>  <DueDateAmount>2017-07-08</DueDateAmount>  <TotalDueAmount>1000</TotalDueAmount>  <NewBalance>10000</NewBalance>  <LastBalance>9000</LastBalance>  <MinimumPaymentAmount>50</MinimumPaymentAmount>  <Transactions>  <Transaction>  <TransactionDate>02-02-2017</TransactionDate>  <TransactionAmount>214.25</TransactionAmount>  <MerchantName>compu me</MerchantName>  <Currency>818</Currency>  <TransactionDescription>Withdrawal</TransactionDescription>  <LedgerEffect>Debit</LedgerEffect>  </Transaction>  </Transactions>  </Body>  </Data>  </CMS.Inquire.Credit.Card.LastStatement.Response>  </message> |

**Sample 2 (Success with No Data Found):**

|  |
| --- |
| <message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Inquire.Credit.Card.LastStatement.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>no data found</ResponseDescription>  </Body>  </Data>  </CMS.Inquire.Credit.Card.LastStatement.Response>  </message> |

**Sample 3 (Failed):**

|  |
| --- |
| <message>  <signature>signature here</signature>  <CMS.Inquire.Credit.Card.LastStatement.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>31002</ResponseCode>  <ResponseDescription>Card not found</ResponseDescription>  </Body>  </Data>  </CMS.Inquire.Credit.Card.LastStatement.Response>  </message> |

* + - * 1. **Transfer To Credit Card Operation**

Avails Transfer to credit card in CMS

**TransferToCreditCardRequest**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClinetId |  | Guid | 36 - 36 | 1 - 1 | * Give n by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Created by sender * Must be unique among the whole CMS requests * Formatted as GUID |
| Channel |  | String | 6-8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| CardNumber |  | String | 16 - 19 | 1 – 1 | * Card number of the primary credit card |
| Amount |  | Decimal | 16 , 2 | 1 – 1 | * The amount of transfer |
| Currency |  | String | 3 - 3 | 1 - 1 | * (818,840) allowed values |
| RRN |  | String | 1 - 12 | 1 - 1 | * Retrieval reference number |
| MessageDescription |  | String | 1 - 25 | 1 - 1 | * Message description for transfer transaction |
| Terminal ID |  | String | 1 - 16 | 1 - 1 | * Terminal id |
| Terminal Location |  | String | 1 - 40 | 1 - 1 | * Terminal location |
| LocalDate |  | DateTime (UTC ) |  | 1 - 1 | * Local transaction date and time |

**Sample:**

|  |
| --- |
| <Message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Transfer.To.Credit.Cards.Request>  <Data>  <Header>  <Version>1.00</Version>  <ClientID>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientID>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <Cardnumber>4532451629337241</Cardnumber>  <Amount>300</Amount>  <Currency>840</Currency>  <RPN>77556533</RPN>  <MessageDescription>a</MessageDescription>  <TerminalID>Term665</TerminalID>  <TerminalLocation>Aswan</TerminalLocation>  <LocalDate>2001-12-17T09:30:47Z</LocalDate>  </Body>  </Data>  </CMS.Transfer.To.Credit.Cards.Request>  </Message> |

###### TransferToCreditCardresponse

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | **1 – 1** |  |
| Version |  | String | 4 - 5 | 1 - 1 | Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Same as RequestId sent in request |
| Channel |  | String | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| ResponseCode |  | String | 5 – 5 | 1 – 1 | * 5 numbers left padded by zeros |
| ReasonDescription |  | String | 1-1024 | 1 – 1 |  |

**Sample 1 (Success):**

|  |
| --- |
| <Message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Transfer.To.Credit.Cards.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientID>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientID>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  < ResponseDescription >success< ResponseDescription/>  </Body>  </Data>  </CMS.Transfer.To.Credit.Cards.Response>  </Message> |

**Sample 2 (Card not found):**

|  |
| --- |
| <Message>  <signature> [signature here](#_Digital_Signature) </signature>  <CMS.Transfer.To.Credit.Cards.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientID>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientID>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>31002</ResponseCode>  <ResponseDescription> card not found <ResponseDescription/>  </Body>  </Data>  </CMS.Transfer.To.Credit.Cards.Response>  </Message> |

* + - * 1. **Inquire Credit Card Transactions By Interval Operation**

Avails inquire credit card transactions by interval in CMS

**InquireCreditCardTransactionsByIntervalRequest**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClinetId |  | Guid | 36 - 36 | 1 - 1 | * Give n by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Created by sender * Must be unique among the whole CMS requests * Formatted as GUID |
| Channel |  | String | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| CardNumber |  | String | 16 - 19 | 1 - 1 | * Number of credit card   Ex. 5118630102038108 |
| FromDate |  | Date | 10 - 10 | 1 - 1 | * The start date of requested interval |
| ToDate |  | Date | 10 - 10 | 1 - 1 | * The end date of the requested interval |

**Sample:**

*<Message>*

<*Signature*> [signature](#_SIGNING) here <*/*Signature>

<CMS.Inquire.Credit.Card.Transactions.ByInterval.Request>

<Data>

<Header>

<Version>1.00</Version>

<ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>

<RequestId>00000000-0000-0000-0000-000000000000</RequestId>

<Channel>mobile</Channel>

</Header>

<Body>

<CardNumber>05124102154789214</CardNumber>

<FromDate>2017-02-02</FromDate>

<ToDate>2017-03-02</ToDate>

</Body>

</Data>

</CMS.Inquire.Credit.Card.Transactions.ByInterval.Request>

*</Message>*

**InquireCreditCardTransactionsByIntervalResponse**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Same as RequestId sent in request |
| Channel |  | String | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| ResponseCode |  | String | 5 – 5 | 1 – 1 | * 5 numbers left padded by zeros |
| ReasonDescription |  | String | 1 - 1024 | 1 – 1 |  |
| Transactions |  |  |  | 0-1 |  |
| Transaction |  |  |  | 1 – \* |  |
| TransactionDate |  | Date | 10 – 10 | 1 – 1 | * Date of the transaction |
| TransactionAmount |  | Decimal | 16 , 2 | 1 – 1 | * The Mount of the transaction |
| MerchantName |  | String | 1 – 25 | 1 – 1 | * The Name of The Merchant with who this transaction was made |
| Currency |  | String | 3 – 3 | 1 – 1 | * The transaction currency   Allowed values 🡺 818 , 840 |
| TransactionDescription |  | String | 1 – 25 | 1 – 1 | * The message description for the transaction |
| LedgerEffect |  | String | 5 – 6 | 1 – 1 | * The transaction effect type   Allowed Values 🡺 Credit, debit |
| TransactionStatus |  | String | 4 – 7 | 1 – 1 | * Allowed values 🡺 Cleared , Hold |

**Sample 1 (Success with Transactions):**

|  |
| --- |
| <message>  <signature>[signature here](#_Digital_Signature)</signature>  <CMS.Inquire.Credit.Card.Transactions.ByInterval.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>Success</ResponseDescription>  <Transactions>  <Transaction>  <TransactionDate>02-02-2017</TransactionDate>  <TransactionAmount>214.25</TransactionAmount>  <MerchantName>name Here</MerchantName>  <Currency>818</Currency>  <TransactionDescription>Withdrawal</TransactionDescription>  <LedgerEffect>Credit</LedgerEffect>  <TransactionStatus>Cleared</TransactionStatus>  </Transaction>  </Transactions>  </Body>  </Data>  </CMS.Inquire.Credit.Card.Transactions.ByInterval.Response >  </message> |

**Sample 2 (Success With no transactions):**

|  |
| --- |
| <message>  <signature>[signature here](#_Digital_Signature)</signature>  <CMS.Inquire.Credit.Card.Transactions.ByInterval.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>no transactions found</ResponseDescription>  </Body>  </Data>  </CMS.Inquire.Credit.Card.Transactions.ByInterval.Response>  </message> |

**Sample 3 (Failed):**

|  |
| --- |
| <message>  <signature>[signature here](#_Digital_Signature)</signature>  <CMS.Inquire.Credit.Card.Transactions.ByInterval.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>31002</ResponseCode>  <ResponseDescription>Card not found</ResponseDescription>  </Body>  </Data>  </CMS.Inquire.Credit.Card.Transactions.ByInterval.Response>  </message> |

* + - * 1. **Inquire Credit Card Transactions By Count Operation**

Avails inquire credit card transactions by count in CMS

**InquireCreditCardTransactionsByCountRequest**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | * Message Version from 1.00 to 99.99 |
| ClinetId |  | Guid | 36 - 36 | 1 - 1 | * Give n by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Created by sender * Must be unique among the whole CMS requests * Formatted as GUID |
| Channel |  | String | 6 - 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| CardNumber |  | String | 16-19 | 1 - 1 | * Number of credit card   Ex. 5118630102038108 |
| Count |  | Integer | 5 - 5 | 1 - 1 | * Count of latest transactions by credit card that will inquire on it |

**Sample:**

*<Message>*

<*Signature*> [signature](#_SIGNING) here <*/*Signature>

<CMS.Inquire.Credit.Card.Transactions.ByCount.Request>

<Data>

<Header>

<Version>1.00</Version>

<ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId >

<RequestId>00000000-0000-0000-0000-000000000000</RequestId>

<Channel>mobile</Channel>

</Header>

<Body>

<CardNumber>05124102154789214</CardNumber>

<Count>00010</Count>

</Body>

</Data>

</CMS.Inquire.Credit.Card.Transactions.ByCount.Request>

*</Message>*

**InquireCreditCardTransactionsByCountResponse**

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Message Item** | **Or** | **Xml Type** | **Min – Max length** | **Min – Max Occurrence** | **Comments** |
| **Header** |  |  |  | 1 – 1 |  |
| Version |  | String | 4 - 5 | 1 - 1 | Message Version from 1.00 to 99.99 |
| ClientId |  | Guid | 36 - 36 | 1 - 1 | * Given by CMS to every client |
| RequestId |  | Guid | 36 - 36 | 1 - 1 | * Same as RequestId sent in request |
| Channel |  | string | 6 – 8 | 1 - 1 | * Channel that bank will send the request from it  ex. internet , mobile |
| **Body** |  |  |  | 1 – 1 |  |
| ResponseCode |  | String | 5 – 5 | 1 – 1 | * 5 numbers left padded by zeros |
| ReasonDescription |  | String | 1 - 1024 | 1 - 1 |  |
| Transactions |  |  |  | 0 - 1 |  |
| Transaction |  |  |  | 1 - \* |  |
| TransactionDate |  | Date | 10 - 10 | 1 - 1 | * Date of the transaction |
| TransactionAmount |  | Decimal | 16 , 2 | 1 - 1 | * The Mount of the transaction |
| MerchantName |  | String | 1 - 25 | 1 - 1 | * The Name of The Merchant with who this transaction was made |
| Currency |  | String | 3 - 3 | 1 - 1 | * The transaction currency   Allowed values 🡺 818 , 840 |
| TransactionDescription |  | String | 1 - 25 | 1 - 1 | * The message description for the transaction |
| LedgerEffect |  | String | 5 - 6 | 1 - 1 | * The transaction effect type   Allowed Values 🡺 Credit, debit |
| TransactionStatus |  | String | 4 - 7 | 1 - 1 | * Allowed values 🡺 Cleared , Hold |

**Sample 1 (Success with Transactions):**

|  |
| --- |
| <message>  <signature>[signature here](#_Digital_Signature)</signature>  <CMS.Inquire.Credit.Card.Transactions.ByCount.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>Success</ResponseDescription>  <Transactions>  <Transaction>  <TransactionDate>02-02-2017</TransactionDate>  <TransactionAmount>214.25</TransactionAmount>  <MerchantName>name Here</MerchantName>  <Currency>818</Currency>  <TransactionDescription>Withdrawal</TransactionDescription>  <LedgerEffect>Credit</LedgerEffect>  <TransactionStatus>Cleared</TransactionStatus>  </Transaction>  </Transactions>  </Body>  </Data>  </CMS.Inquire.Credit.Card.Transactions.ByCount.Response>  </message> |

**Sample 2 (Success With no transactions):**

|  |
| --- |
| <message>  <signature>[signature here](#_Digital_Signature)</signature>  <CMS.Inquire.Credit.Card.Transactions.ByCount.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>00000</ResponseCode>  <ResponseDescription>no transactions found</ResponseDescription>  </Body>  </Data>  </CMS.Inquire.Credit.Card.Transactions.ByCount.Response>  </message> |

**Sample 3 (Failed):**

|  |
| --- |
| <message>  <signature>[signature here](#_Digital_Signature)</signature>  <CMS.Inquire.Credit.Card.Transactions.ByCount.Response>  <Data>  <Header>  <Version>1.00</Version>  <ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>  <Channel>mobile</Channel>  </Header>  <Body>  <ResponseCode>31002</ResponseCode>  <ResponseDescription>Card not found</ResponseDescription>  </Body>  </Data>  </CMS.Inquire.Credit.Card.Transactions.ByCount.Response>  </message> |

# Security



## Authentication

Authenticating request/response for each service is required. The sender must provide request node containing the digital signature for each request sent to CMS. The signature is calculated on the whole request document. Also the receiver must provide response node containing the digital signature for each response sent to CMS.

The resulting Signature node should then be appended to the previously existing request/response parent node of the message.

The used Certificate is X509, in cryptography X.509 is an important standard for a public key infrastructure (PKI) to manage digital certificates and public-key encryption and a key part of the Transport Layer Security protocol used to secure web services.

## Digital Signature

The following table identifies the various algorithms used with the XML digital signing specification. Algorithms are recognized by URIs that exist as an attribute to the element that identifies the algorithms role (DigestMethod, Transform, SignatureMethod, or CanonicalizationMethod).

|  |  |
| --- | --- |
| Methods | Algorithm |
| CanonicalizationMethod | "http://www.w3.org/TR/2001/REC-xml-c14n-20010315" |
| SignatureMethod | "http://www.w3.org/2000/09/xmldsig#rsa-sha1" |
| Transform | http://www.w3.org/2000/09/xmldsig#envelopedsignature" |
| DigestMethod | "http://www.w3.org/2000/09/xmldsig#sha1" |

Sample Message Containing Signature

The following sample message contains a digital signature in the Request node.

<Message>

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<SignedInfo>

<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />

<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />

<Reference URI="">

<Transforms>

<Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />

</Transforms>

<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />

<DigestValue>txwKx23hmioxOgn0Zt8ir1JtqYA=</DigestValue>

</Reference>

</SignedInfo> <SignatureValue>QSeHOanYFUm2eRvXGhEDuqqNYG5AxsGF4kI8pWHiveOBq1Yw9NlMXtcaBq70m4u9tXfS2Cpf5+kICbFV+jcFXY4sHZneETyE7PeY0NeLSMHOb0Mt1/KA9h6dPX2JvO9lh68zx4rFLHMKkxes6n1PNfsQfz7dM8Fgqpk//5gS1sc=</SignatureValue>

</Signature>

<CMS.Inquire.Customer.Cards.Request>

<Data>

<Header>

<Version>1.00</Version>

<ClientId>d15a234c-72dd-44d4-8b96-8e4b7d5de81a</ClientId>

<RequestId>00000000-0000-0000-0000-000000000000</RequestId>

</Header>

<Body>

<CustomerId>932432421</CustomerId>

</Body>

</Data>

</CMS.Inquire.Customer.Cards.Request>

</Message>

# Appendixes

## Appendix A

**Card Status**

|  |  |
| --- | --- |
| **Status Id** | **Status Description** |
| 0 | Inactive |
| 1 | Active |
| 2 | Lost Card |
| 3 | Stolen Card |
| 4 | Restricted |
| 5 | VIP |
| 6 | Check reason code field |
| 9 | Closed |

## Appendix B

**General response codes**

|  |  |
| --- | --- |
| **Response Code** | **Response Description** |
| 00000 | Approved Or Completed Successfully |
| 70001 | Invalid XML Message |
| 70002 | Invalid Client Id |
| 70003 | Invalid Message Signature |
| 70004 | Unexpected Message Root |
| 70005 | Invalid Field |
| 99998 | General Error |
| 99999 | System Error |

**CMS response codes**

|  |  |
| --- | --- |
| **Response Code** | **Response Description** |
| 31001 | Customer not found |
| 31002 | Card not found |
| 31003 | Invalid Card Status |