2013

Integration with Merchant's Software System

finaccess
enhance • enable • empower

Ekantakuna, Opposite DFID
GPO Box 2904, Ward No. 13,
Lalitpur, Nepal.
Tel: +977 1 5548016 / 5548017, Fax: +977 1 5548116
www.hellopaisa.com.np

Confidentiality

This volume is a product of **finaccess**!

This information is provided on the strict understanding that it is confidential and must be treated by you as such.

The information contained in this document may only be divulged to those members of your staff on a need to know basis and you must ensure that they are made aware of the confidential nature of the information.

You must also use all reasonable endeavors to ensure they keep such information confidential.

Non-Disclosure Agreement

The reader of this document hereby undertakes to keep all information, documents, and knowledge of whatever kind obtained by them or by any of our advisers or agents (whether orally, in writing or in any way recorded or by observation (—the Information), **finaccess** Pvt Ltd in connection with the proposals referred to in clause three below (hereafter referred to as — **finaccess**) secret and confidential and we further undertake, agree and acknowledge the following:

i. The reader will not (save with the prior written consent of **finaccess** or as required by law or the relevant regulatory body) communicate any of the Information to any other person or body, (—except associated companies), in connection with the proposals referred to in the clause below.

ii. The reader will not make or permit or procure to be made or solicit or assist any other person to make any announcement or disclosure of our discussions with **finaccess**, whether or not any discussions commence, proceed or are terminated, without **finaccess**'s prior written consent except as required by law or competent regulatory body. If agreements to any cooperation with **finaccess**, at or after the date of such cooperation, no announcement of the cooperation will be made except by mutual agreement between **finaccess** and the reader or as required by law or competent regulatory body.

iii. The reader will make no use of the Information for any purpose other than giving consideration to the possibility of putting forward proposals concerning a potential cooperation with **finaccess** or for the purpose of such proposals, and, in particular, but without prejudice to the generality of the foregoing, the reader will not use the Information to obtain any commercial benefit or for any purpose detrimental to **finaccess**.

Version

Version	Date	Author / Initials	Commentary
1.4	07 October 2013	Bikash Saran	Updated with report section and registration API

Table of Contents

A) **Purpose** 4

B) **Specification of Data Exchange Specification** 5

C) **Web Service Class Diagram** 6

D) **Payment Request** 7

E) **Status API** 8

F) **Customer Registration API** 9

G) **OTP Verification API** 9

H) **Bank Account Integration API** 10

I) **Cash In API** 10

J) **Cash Out API** 10

K) **IP Address & URL and Transaction Reports** 13

L) **Contact Details** 13

Cell-O^{r} Payment API

A) Purpose

The purpose of this document is to describe the possibilities of integrating the Hello Paisa Mobile Financial Services platform with merchant/3rd party software systems in order to implement the mechanism of payment reception by the merchants/3rd parties in favor of customer that are consuming the merchant/3rd party's services.

B) Specification of Data Exchange Specification

Hello Paisa provides a set of web-methods for merchant/3rd party software system to communicate with in order to register a payment request and verify the requested payment.

Execution of payment request consists of 2 stages:

- **InitiatePayment** method should be used for payment operation initialization. It checks whether a given mobile wallet [account] is valid and can be debited. After successful verification the transaction will go through.

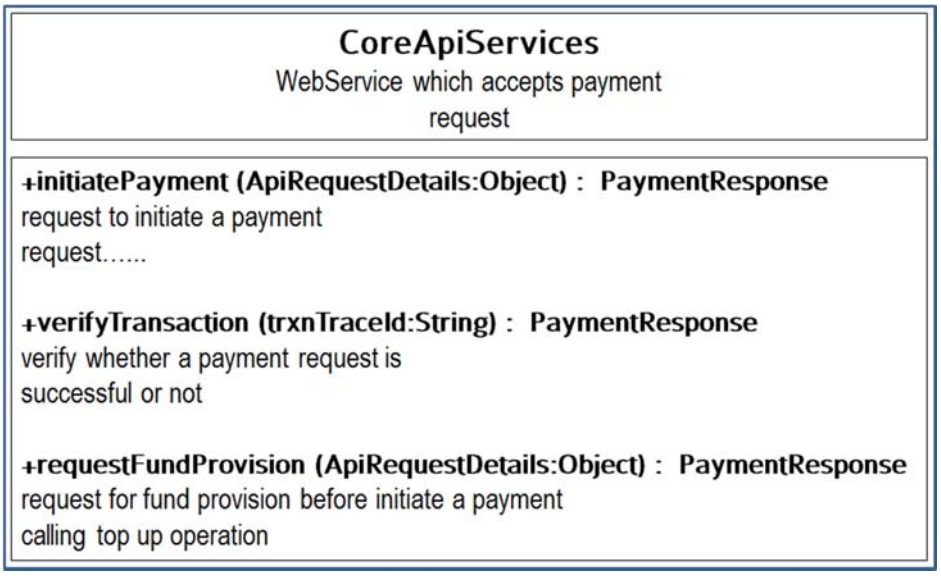
- And **VerifyTransaction** needs to be called for payment confirmation whether success or fail.

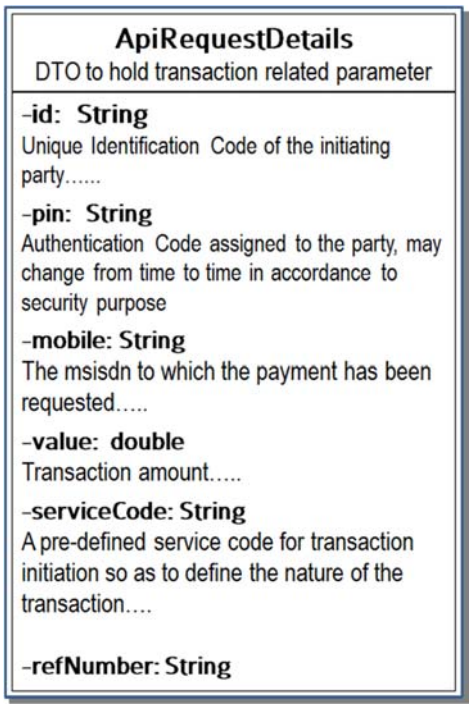
- **IntiateRegistrationRequest** method should be used for registering a potential Hello Paisa customer. On successful acceptance of the request by the Cell-O r system, an OTP [One Time Passcode] is sent to the intended customer on his/her cellphone.

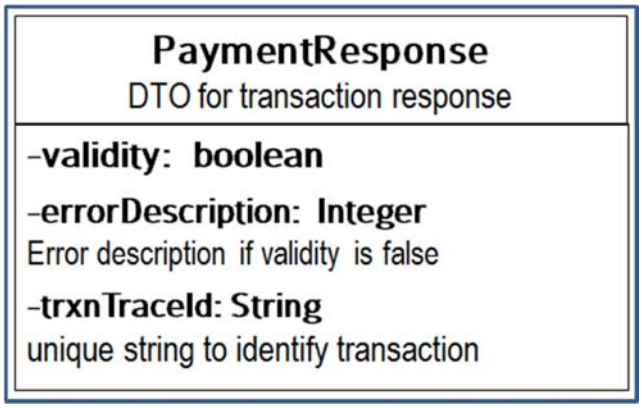
- **VerifyOtpRequest** method is called with the OTP after the intended customer has received it. On successful acceptance of the request, an IVR call will be generated to the customer for setting a 4-digit Pin [which is a password to authenticate any transaction in future over IVR].

- **InitiateBankLinkRequest** method is called to link a bank account to a customer's SVA. The linked bank account is further verified and approved by the bank.

C) Web Service Class Diagram







D) Payment Request

Web method initiatePayment() can be called to initiate payment requests.

Sample Java Code as follows:

```
CoreApiServicesClient apiObject = new CoreApiServicesClient();
CoreApiClient.paymentResponse response = new CoreApiClient.paymentResponse();
ApiRequestDetails request = new ApiRequestDetails();
request.setId = "0000001346"; //This is ID for test
request.setPin = "A6E8ED8F8888503B";   //This is PIN for test
request.setServiceCode = "23";  //This is PIN for test
request.setValue = 2000.0;
response = apiObject.initiatePayment(request);
```

E) Status API

The response code 9003, 9004 and 9006 reflects that the transaction is still in progress and hence, the transaction should be considered as "in progress" and not a successful one or the failed one.

Sample Java Code as follows:

```
CoreApiServicesClient apiObject = new CoreApiServicesClient();
CoreApiClient.paymentResponse response = new CoreApiClient.paymentResponse();
ApiRequestDetails request = new ApiRequestDetails ();
request.setId = "0000001346";
request.setPin = "A6E8ED8F8888503B";
request.setTrxnTraceId = "XXXXXXXXXX";
response = apiObject.verifyTransaction(request);
```

Alternately, the Cell-O system is able to call a post back method on the merchant/3rd party's side confirming a successful transaction. For this to be enabled, the merchant/3rd party has to provide the mechanism to call the function on their side.

F) Customer Registration Request API

Web Method initiateRegistrationRequest() can be used to initiate a registration process of a potential customer.

Sample Java Code as follows:

```
CoreApiServicesClient apiObject = new CoreApiServicesClient();
CoreApiClient.paymentResponse response = new CoreApiClient.paymentResponse();
ApiRegistrationRequestDetails request = new ApiRegistrationRequestDetails ();
request.setId = "0000001346";
request.setPin = "A6E8ED8F8888503B";
request.setServiceCode = "106";
request.setFirstName= "Ram Bahadur";
request.setLastName = "Thapa";
request.setDob = "01011990";
request.setDobType= "a";
request.setLanguagePreference = "n"; [n for Nepali, e for English]
request.setCustomerIdNo = "12345/123";
request.setCustomerIdType = "1"; [1 for Citizenship Number, 2 for others]
request.setCustomerProfileNo = "123"; [Product Id as provided by the bank]
request.setMobileNo= "9851111111";

response = apiObject.initiateReqistrationRequest(request);
```

G) Verify OTP Request API

Web Method initiateVerifyOtpRequest() can be used to authorize the registration process of a customer. The customer needs to pass the OTP received on his/her mobile no, which the initiating party will send along with the customer's mobile number over API method call.

Sample Java Code as follows:

```
CoreApiServicesClient apiObject = new CoreApiServicesClient();
CoreApiClient.paymentResponse response = new CoreApiClient.paymentResponse();
ApiVerifyOtpRequestDetails request = new ApiVerifyOtpRequestDetails ();
request.setId = "0000001346";
request.setPin = "A6E8ED8F8888503B";
request.setServiceCode = "97";
request.setOtp=″1234″;
request.setMobileNo= "9851111111";

response = apiObject.initiateVerifyOtpReqistrationRequest(request);
```

H) Bank Account Integration API

Web method initiateBankLinkRequest() can be called to link a bank account to a customer's SVA.

Sample Java Code as follows:

```
CoreApiServicesClient apiObject = new CoreApiServicesClient();
int response;
response = apiObject.initiateBankLinkRequest(mobileNo, bankAccount);
```

I) Cash In Request

Web method initiatePayment() can be called to initiate cashin requests.

Sample Java Code as follows:

```
CoreApiServicesClient apiObject = new CoreApiServicesClient();
CoreApiClient.paymentResponse response = new CoreApiClient.paymentResponse();
ApiRequestDetails request = new ApiRequestDetails();
request.setId = "0000001346"; //This is ID for test
request.setPin = "A6E8ED8F8888503B";   //This is PIN for test
request.setServiceCode = "113";              //This is Service Code for CashIn
request.setValue = 2000.0;
response = apiObject.initiatePayment(request);
```

J) Cash Out Request

Web method initiatePayment() can be called to initiate cashout requests.

Sample Java Code as follows:

```
CoreApiServicesClient apiObject = new CoreApiServicesClient();
CoreApiClient.paymentResponse response = new CoreApiClient.paymentResponse();
ApiRequestDetails request = new ApiRequestDetails();
request.setId = "0000001346"; //This is ID for test
request.setPin = "A6E8ED8F8888503B";   //This is PIN for test
request.setServiceCode = "12";  //This is Service Code for Cashout
request.setValue = 2000.0;
response = apiObject.initiatePayment(request);
```

Result: PaymentResponse [Also refer the class diagram above in Section: A]

a. *Boolean: validity*
   Request Successful: TRUE
   Request Failed: FALSE

b. *Integer: errorDescription*
   Error codes as in the following table (next page):

ERROR CODES Table

Error Code	Error Type	Error Code	Error Type
101	SENDER_DOES_NOT_EXIST	231	BANK_ACCOUNT_INVALID
102	SENDER_PROFILE_NOT_EXIST	232	BANK_ACCOUNT_VALIDATION_FAILED
103	SENDER_SERVICE_NOT_ASSIGNED	281	PROFILE_NOT_INSERTED
104	SENDER_INACTIVE_STATUS	301	FEE_NOT_FOUND
105	SAME_SENDER_RECEIVER	302	MULTIPLE_FEE_FOUND
106	SENDER_CUSTOMER_BLACKLIST	303	FEE_CALCULATION_ERROR
107	SENDER_BANK_ACCOUNT_NOT_EXISTING	304	BALANCE_NOT_FOUND
108	SENDER_PROFILE_NOT_ACTIVE	305	MULTIPLE_BALANCE_FOUND
112	RECEIVER_PROFILE_NOT_EXIST	306	INSUFFICIENT_BALANCE
114	RECEIVER_INACTIVE_STATUS	311	TRANSACTION_REQUEST_DUMP_ERROR
115	SAME_RECEIVER_SENDER	321	LIMIT_NOT_FOUND
116	RECEIVEFR_CUSTOMER_BLACKLIST	322	PER_TRANSACTION_LIMIT_EXCEEDS
117	RECEIVER_BANK_ACCOUNT_NOT_EXISTING	323	DAILY_LIMIT_EXCEEDS
118	RECEIVER_PROFILE_NOT_ACTIVE	324	MONTHLY_LIMIT_EXCEEDS
119	RECEIVER_UPAY_ALREADY_AVAILABLE	325	HALF_YEARLY_LIMIT_EXCEEDS
151	USER_NOT_EXISTS	326	YEARLY_LIMIT_EXCEEDS
152	MULTIPLE_USERS_FOUND	327	LIMIT_DUPLICATE_ERROR
153	USER_NOT_ACTIVE	351	AUTHORIZATION_FAILED
154	WRONG_PASSWORD	361	ISO_ERROR
156	USER_ALREADY_LOGGED_IN	401	PIN_NOT_FOUND
157	USER_DOES_NOT_HAVE_ANY_ROLES	402	MULTIPLE_PIN_FOUND
200	INVALID_SERVICE	403	PIN_COUNT_RESET_ERROR
201	KEYWORD_INVALID	404	WRONG_PIN_MULTIPLE_TIMES
202	KEYWORD_FULL_PATTERN_INCORECT	405	ENTERED_PIN_NOT_MATCHED
203	MOBILENO_NO_DIGIT	406	NEW_PIN_NOT_TRAPPED_CORRECTLY
204	MOBILENO_INCORRECT	461	OUT_MESSAGE_DUMP_ERROR

205	AMOUNT_INCORRECT	501	NO_PIN_CAPTURE
206	AMOUNT_EXCEEDED	502	PREMATURE_HUNGUP
207	PIN_FORMAT_ERROR	503	USER_HUNGUP
208	BANK_ACCOUNT_CONTAINS_OTHER_THAN_ALPHANUMERIC	504	ALL_CALL_RETRY_FAIILED
211	KEYWORD_UNDEFINED_ERROR	505	TRANSACTION_CANCELLED_BY_USER
212	AMOUNT_ZERO_OR_LESS_ERROR	601	NO_RECHARGE_VOUCHER_FOUND
213	REGISTRATION_INCORRECT_DATETYPE	901	EMPTY_RESULT_DATA_ACCESS_EXCEPTION
214	REGISTRATION_SYSTEM_NOT_SUPPORTED_IVR_LANGUAGE	902	NO_AMI_CONNECTION_DETAILS_FOUND_IN_DATABASE
215	REGISTRATION_INCORRECT_DATE_FORMAT	903	DB_ACCESS_ERROR
216	REGISTRATION_INCORRECT_DATE_LENGTH	951	SOCKET_NOT_CONNECTED
217	FIRST_NAME_INCORRECT	952	SOCKET_SENDRECEIVE_ERROR
218	LAST_NAME_INCORRECT	953	SOCKET_TIME_OUT_ERROR
219	INVALID_IDENTIFICATION_DOC_TYPE	960	SOCKET_UNKNOWN_ERROR
220	INVALID_IDENTIFICATION_DOC	999	UNKNOWN_ERROR
221	IDENTIFICATION_NUMBER_MISSING	9001	FAILED_AUTHENTICATION
222	OTP_INCORRECT_FORMAT	9002	SUCCESSFUL_AUTHENTICATION
223	DOB_UNDER_REQUIRED_AGE	9003	RETRYING_CALL
224	REGISTRATION_CUSTOMER_DIED	9004	CALL_IN_PROGRESS
225	REGISTRATION_CUSTOMER_YET_TO_BORN	9005	CALL_NOT_QUEUED
226	REGISTRATION_NEPALI_DOB_INVALID_DAY_OF_MONTH	9006	CALL_QUEUED
227	REGISTRATION_PROFILE_NOT_FOUND	9006	AUTHENTICATION_NOT_COMPLETE
506	**TELCO_NOT_SUPPORTED**	**507**	**NO_ANSWER**
508	**MULTIPLE_TRANSACTION_AUTHENTICATION_CALLED**		

Note: Not all error codes may be in use the API. Error Code:0 represents successful transaction.

c. *String: trxnTraceId*
Transaction Id

K) IP Address & URL and Transaction Reports

The web service definition language (WSDL) URL for the payment API for test platform is http://110.44.114.210:8080/PaymentHandler/coreApi?wsdl.

Similarly, the merchant/3rd party's IP address has to be provided so that the required access policies can be added at network level for the inter-connection.

There will be an additional web portal at https://report.hellopaisa.com.np/ available for merchants/3rd party from where they can view report/statement, check and verify the transactions. This portal will be accessible only after the merchant or 3rd party has been registered in the Hello Paisa live system by one of the partner banks.

On test platform, this will be enabled at http://110.44.114.210:8082 after we provide the credentials of a registered merchant for testing purpose.

L) Contact Details

For more details and queries, please contact:

Bikash Saran, Lead Developer/Analyst
bikash.saran@finaccess.com.np
Tel: 977-1-5548016, 977-1-5548017.

dipak.shrestha@finaccess.com.np
Dipak Shrestha, Chief Technology Officer
Tel: 977-1-5548016, 977-1-5548017.