SDK Simulator Guide

for Vodafone Fiji

Document Version 1.0.0

Vodafone Fiji Limited

168 Princess Road

Tamavua, Suva

Fiji Islands

Phone: (679) 331 2000

Fax: (679) 331 2007

Document Code VDF-SDK-UGD v1.0.0 Last edited: 28 December 2013

Table of contents

1 Overview 6

2 Prerequisites 7

2.1 Start Up the Simulator 7

2.2 Stop the Simulator 7

3 Using the Simulator 8

3.1.1 SMS MT 8

3.1.2 SMS MO 9

3.1.3 USSD MT 15

3.1.4 USSD MO 16

3.1.5 CAAS Request 18

3.1.6 Response Error Codes 26

Change Control

Version	Date	Description	Author
0.0.1	11/07/2013	Updated SDK Simulator Guide for Vodafone Fiji	Kalpanie Ratnayake
0.2.0	12/07/2013	Reviewed Document	Roshni Hewamallika
0.2.1	12/07/2013	Updated document based on comments from Roshni	Kalpanie Ratnayake
0.2.2	27/12/2013	Updated to Vodafone branding colours	Kalpanie Ratnayake
0.2.3	27/12/2013	Vodafone address updated based on comments from Duleepa.	Kalpanie Ratnayake

About this document

The purpose of this hSenid Mobile document is to provide sufficient information on the usage of SDK Simulator for Vodafone, Fiji.

The intended audience for this document is the Content Developers on Vodafone, Fiji.

The document is divided into the following chapters:

Chapter	Description
1 Overview	This chapter gives a brief description of SDK Simulator for Vodafone.
2 Prerequisites	This chapter gives a brief description of Software requirements for Vodafone.
3 Using the Simulator	This chapter gives a brief description of features and usage of the Simulator.

1 Overview

SDK Simulator creates a virtual environment for the Content Providers (CPs) to test the created applications before connecting with the real-world system. The applications can be tested with actual validations using the simulator.

2 Prerequisites

Following are the software requirements to install the SDK Simulator at Vodafone, Fiji.

- Java – version 1.6.0 or upper

- **Install Java in Windows:**

http://java.com/en/download/help/windows_manual_download.xml

- **Install Java in Linux:**

http://www.java.com/en/download/help/linux_install.xml

Other required libraries are bundled in the package.

NOTE: Distribution file will come as a zipped file and extract it to the destination folder.

2.1 Start Up the Simulator

Step 1: Extract the sdk-standalone-1.0.0-distribution.zip file

Step 2: Go to the bin folder of the extracted file

Step 3: Start the SDK using sdp-simulator.bat file placed in the bin folder.

NOTE: If the application is running on Linux, sdp-simulator.sh file should be used.

Linux - sh sdp-simulator console

Windows - sdp-simulator.bat console

SDP SDK can be accessed from URL *http://localhost:10001/mchoice-sdp-sdk*

NOTE: An application with the respective NCS enabled, also has to be run with the simulator.

2.2 Stop the Simulator

Press Ctrl + C to stop the simulator

3 Using the Simulator

SDP clients can send requests to the SDP Simulator. Created applications can be used to send requests to the simulator.

For SMS and USSD services, there are phone simulations and window to send messages; whereas for CAAS the requests and messages sent to application are displayed in tabular form.

Each service will be described in subsequent sections.

3.1.1 SMS MT

Figure 3.1

"**Customers Phone**" displays the messages received to the customer's phone.

When a message is received to the customer's phone, the phone screen displays the message as shown in the figure above.

Similarly, the "**Messages sent to Customer**" section shows the messages with the details (message sent time, destination phone number and message content) as shown in the figure below.

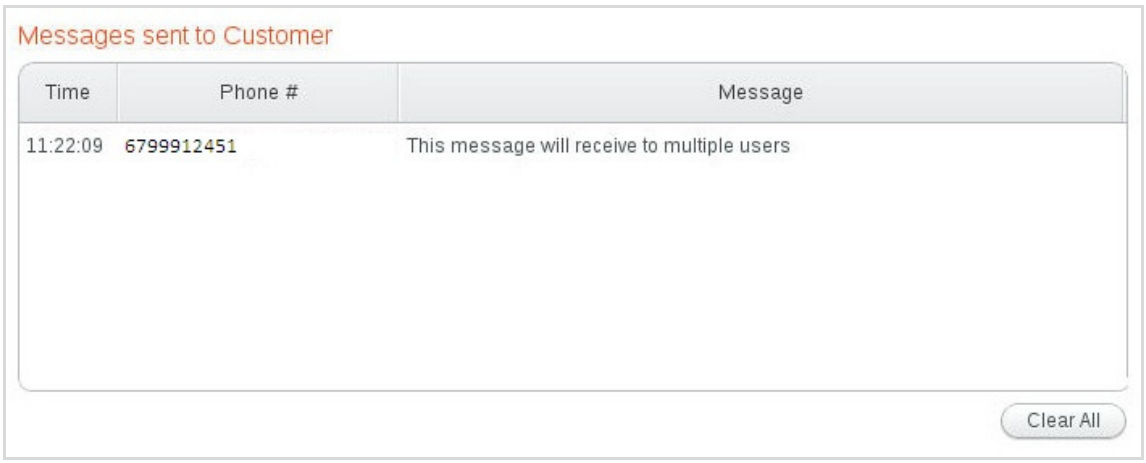

Figure 3.2

User can still test the MT part alone (without MO) with any application provided that it can send MT messages.

For that, send the MT message to the following URL (which accepts SMS MT) and the MT message will be displayed in the simulator.

http://localhost:7000/sms/send

3.1.2 SMS MO

To send a message to the application, the CP can use the provided interface in the simulator. Refer the figure below.

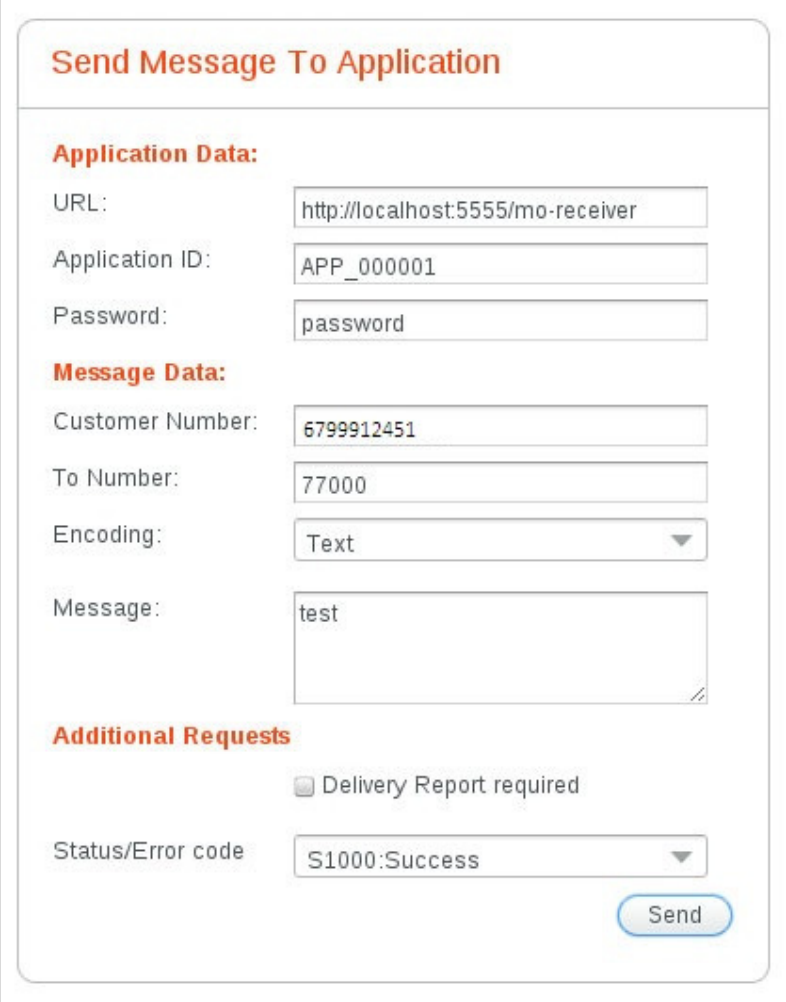

Figure 3.3

Above interface is to simulate message sending to the application.

If Delivery Report is required, the option under "**Additional Requests**" should be selected. Then the "**Delivery Report URL**" has to be specified, as in the figure below.

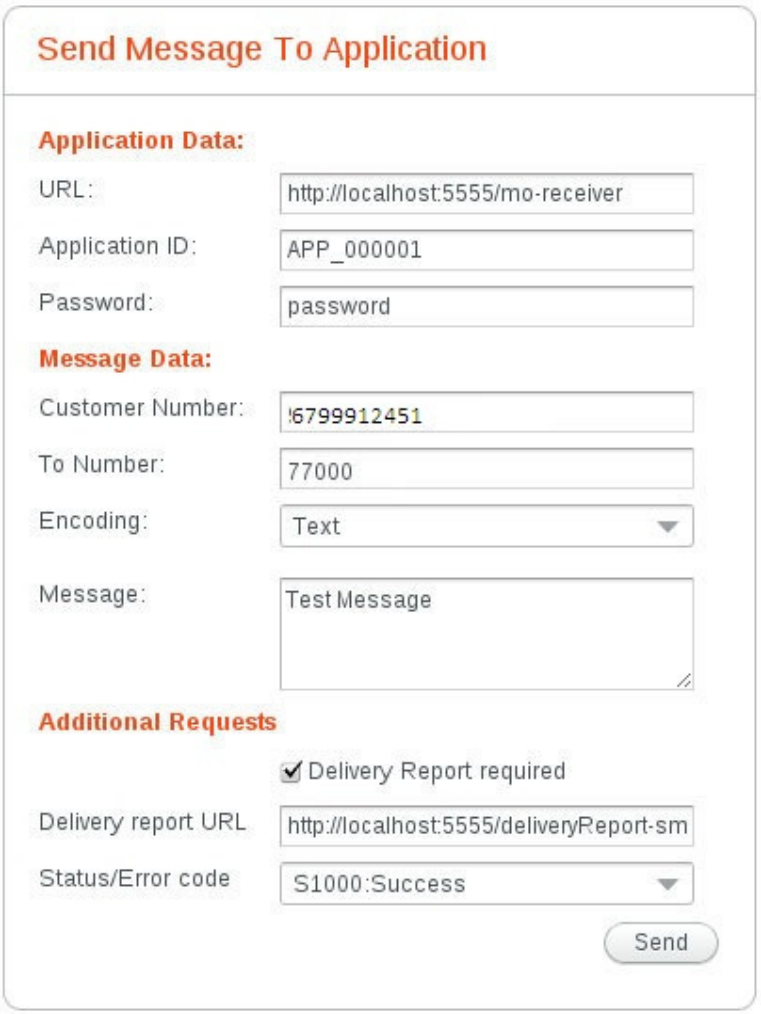

Figure 3.4

Enter the details as follows.

Field name	Description	Sample value
URL	The URL where the application is hosted	http://localhost:5555/mo-receiver
Application ID	Unique ID of the Application	APP_00001
Password	Password of the application. The application password will be auto-generated when the application is created.	password

Customer Number	The phone number from which the message is sent	94779876543
To Number	The phone number which receives the message	7788
Encoding	Message encoding type	Text, Binary (If the encoding type is set to "Binary" the Binary header should be specified. This is applicable only for SMS service. Refer figure 3.5)
Message	Message content	Test Message
Delivery Report Required	Select the check box to generate Delivery Report request from the application side	
Delivery Report URL	URL which sends delivery reports to the application.	http://localhost:5555/deliveryReport-sms
Status/Error Code	Select the response status/error code to generate, when MT (Mobile Termination) message received from application.	S1000:Success

Once the values are entered correctly, click on the "**Send**" button.

Following figure shows sample Status/Error codes.

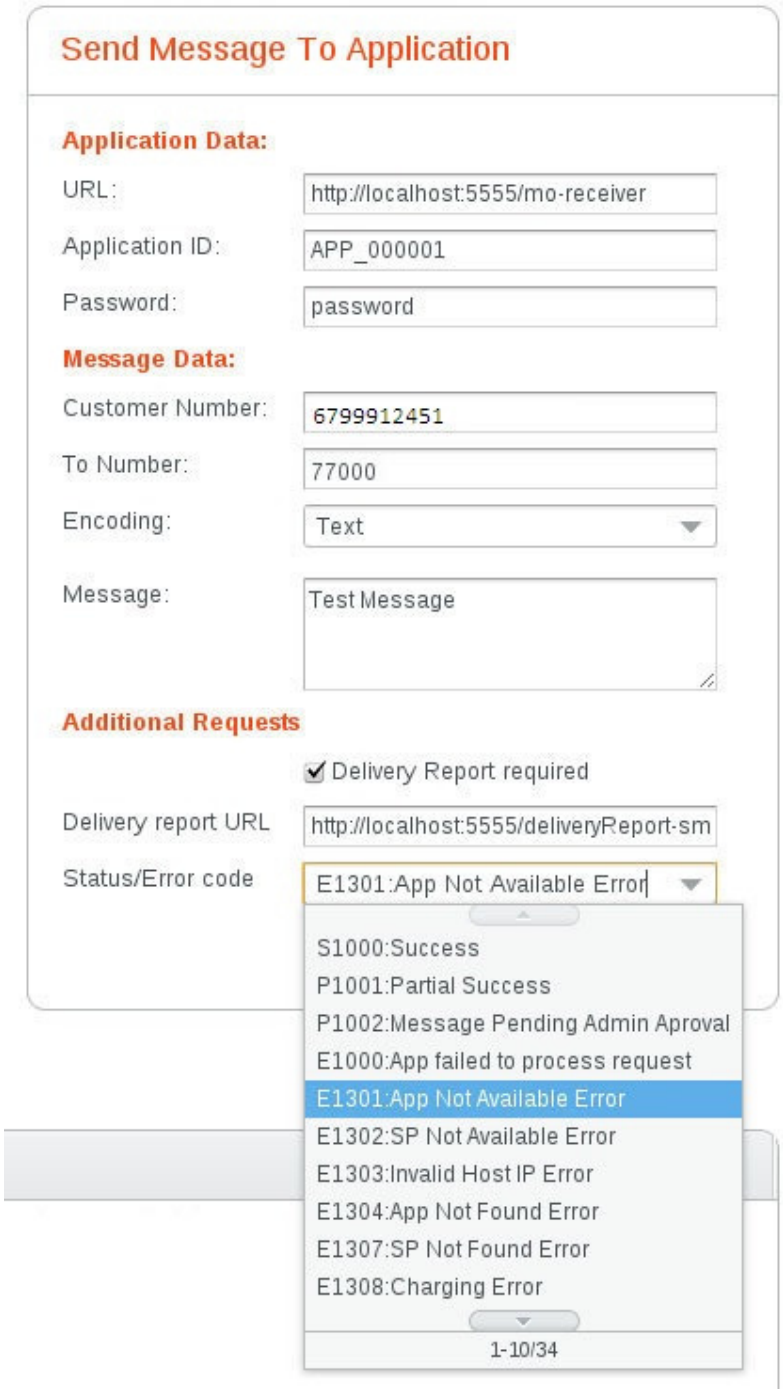

Figure 3.5

If the "**Encoding**" type is selected as "**Binary**", a Binary Header has to be specified. Refer the figure below.

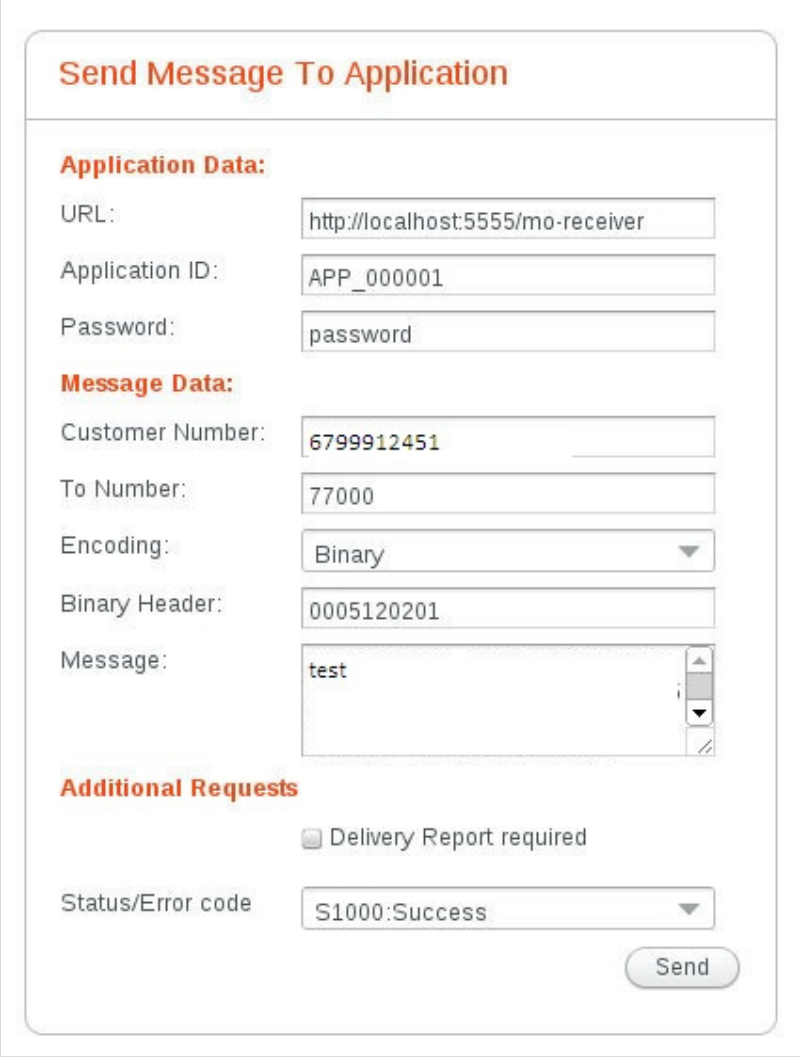

Figure 3.6

Once the values are entered correctly, click on the "**Send**" button.

The messages sent to the application will be displayed with its sent time, phone number, message content and status under "**Message sent to Application**" section.

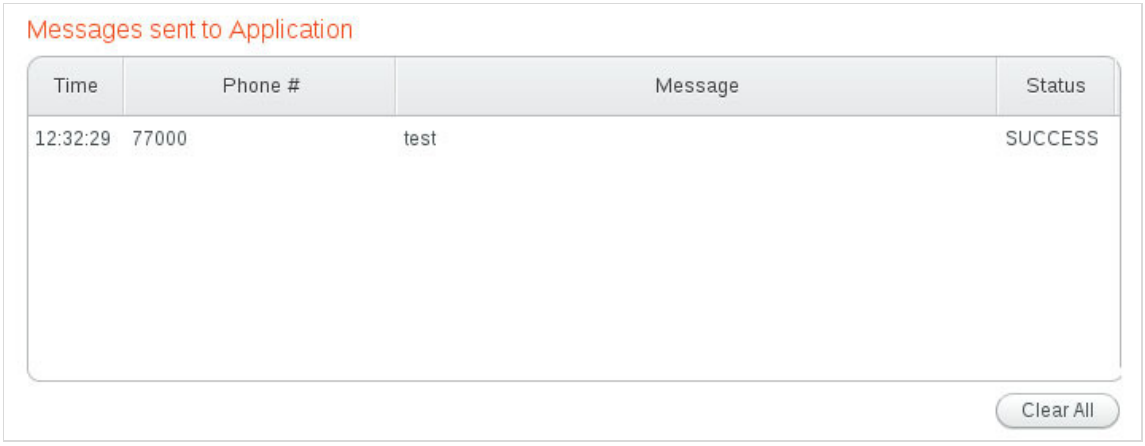

Figure 3.7

3.1.3 USSD MT

Figure 3.8

Once connected to the application, the USSD prompt will be shown with the options.

The "**Messages sent to Customer**" section shows the messages with the details (message sent time, destination phone number and message content) as shown in the figure below.

Figure 3.9

User can still test the MT part alone (without MO) with any application provided that it can send MT messages.

For that, send the MT message to the following URL (which accepts USSD MT) and the MT message will be displayed in the simulator.

http://localhost:7000/ussd/send

3.1.4 USSD MO

To send a message to the application, the CP can use the provided interface in the simulator. Refer the figure below.

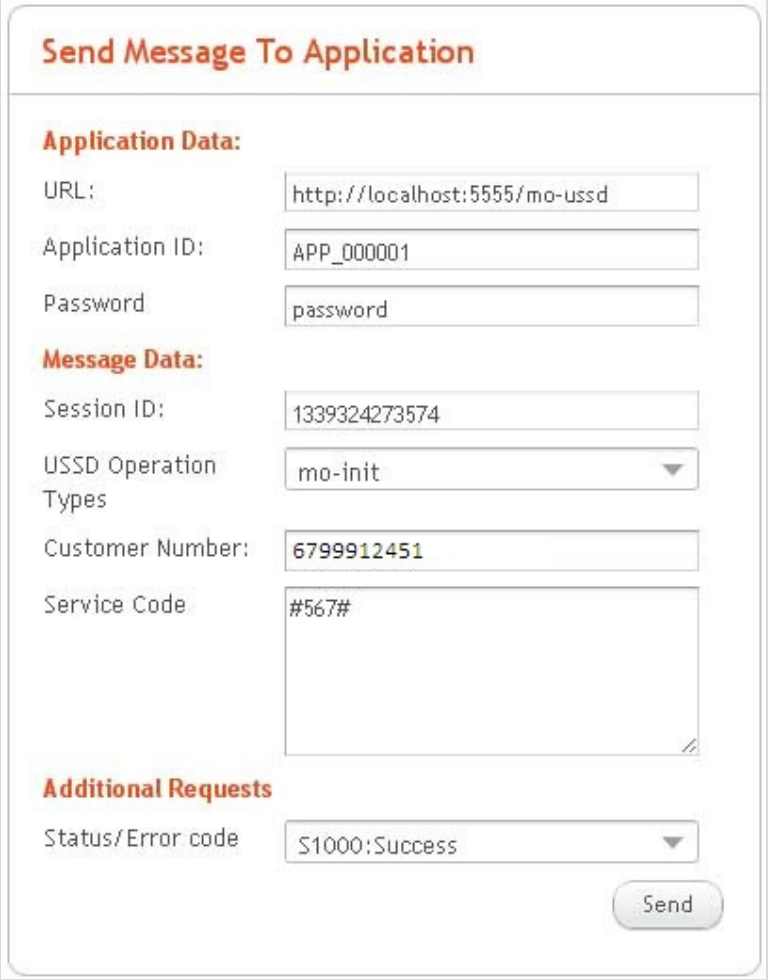

Figure 3.10

Enter the details as follows.

Field name	Description	Sample value
URL	The URL where the application is hosted	http://localhost:5555/mo-ussd
Application ID	Unique ID of the Application	APP_00001
Password	Password of the application. The application password will be auto-generated when the application is created.	password

Customer Number	The phone number from which the message is sent	94775576481
Service Code	USSD Service Code	*141#
Status/Error Code	Select the response status/error code to generate, when MT (Mobile Termination) message received from application.	S1000:Success

Once the values are entered correctly, click on the "**Send**" button.

The messages sent to the application will be displayed with its sent time, phone number, message content and status under "**Message sent to Application**" section.

Figure 3.11

3.1.5 CAAS Request

CaaS (Charging as a Service) enables applications to charge and to get account information from subscribers.

This simulator can set an initial balance for the account. Balance can be queried (in Query Balance) and also can be used in debit transactions (Direct Debit).

3.1.5.1 Query Balance

Application can send the “Query Balance” request to the simulator to retrieve the current balance of the particular subscriber’s account.

The application has to send the query balance request to following url

http://localhost:7000/caas/get/balance

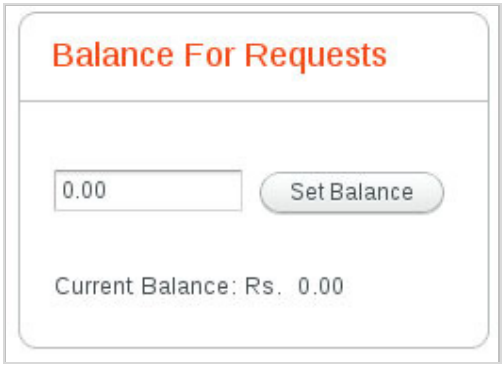

Figure 3.12

To set the balance, user needs to enter the required amount in the relevant text field and click on “Set Balance”. The amount will be set as the account balance.

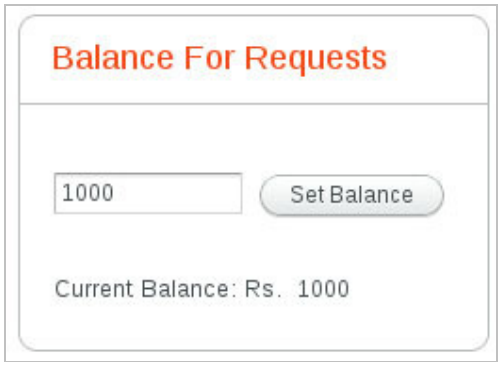

Figure 3.13

In addition, the values set for Account type, Account Status are used in the Query Balance response.

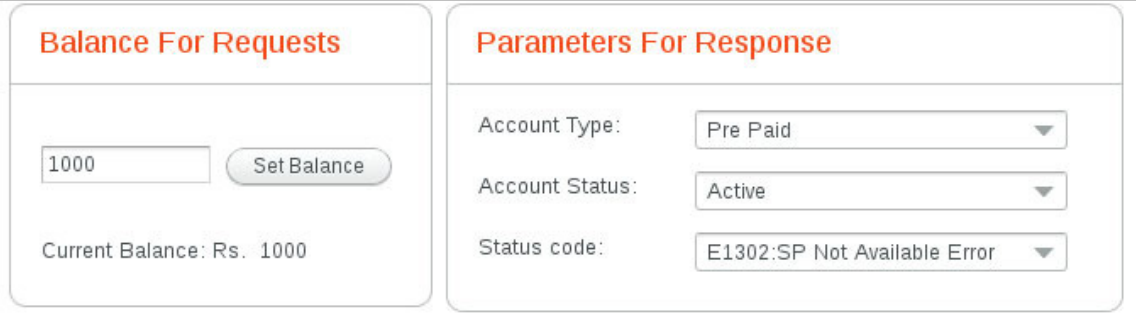

Figure 3.14

To simulate error scenarios, the user has to select the appropriate status code from the "Status Code" list.

Field name	Description	Sample value
Set Balance	Sets the initial Balance for the account.	
Account Type	Select the account type	Pre paid, Post paid
Account Status	Specify the account status	Active, Blocked
Status Code	Set the Status Code	

Refer following for the sample request and response.

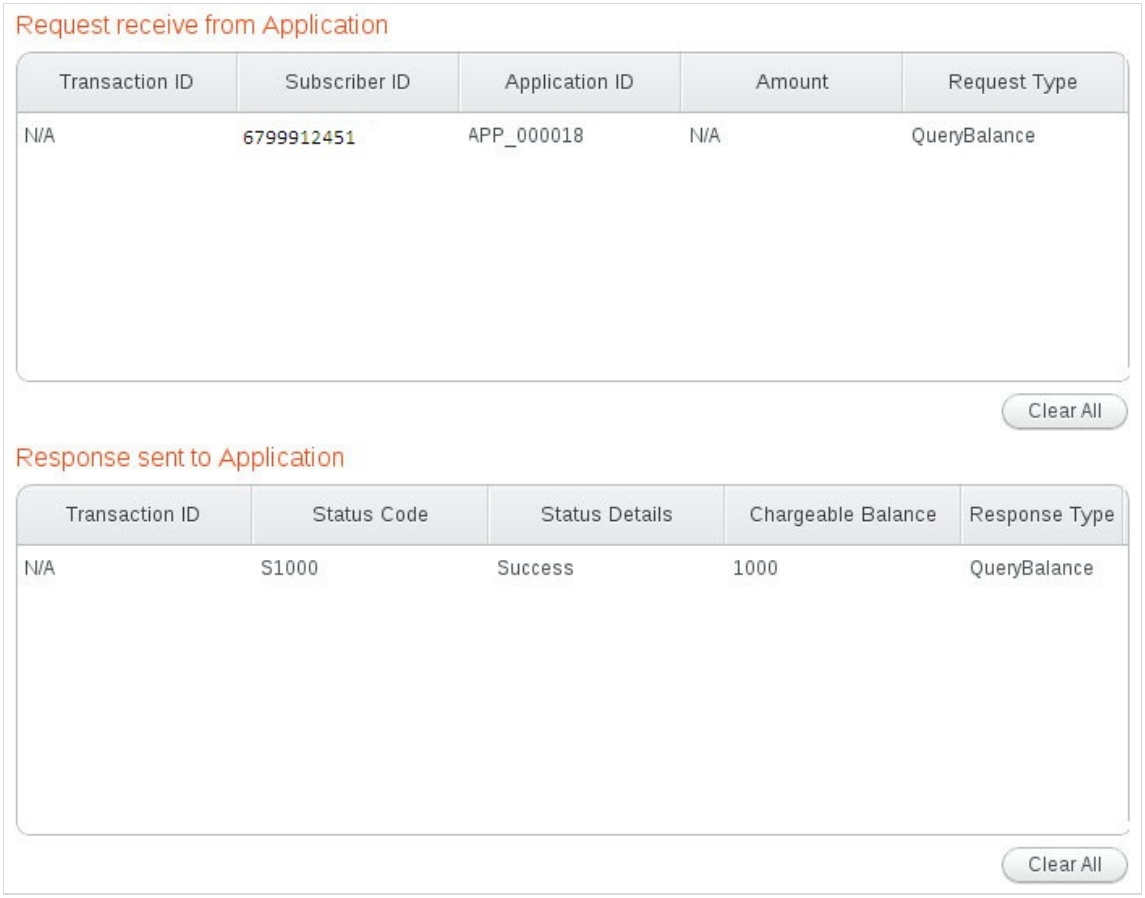

Figure 3.15

When Query Balance Request is sent, the response will send the current account balance along with other account related details. (Refer the figure above.)

In case of an error scenario, refer the following screens for the request and response.

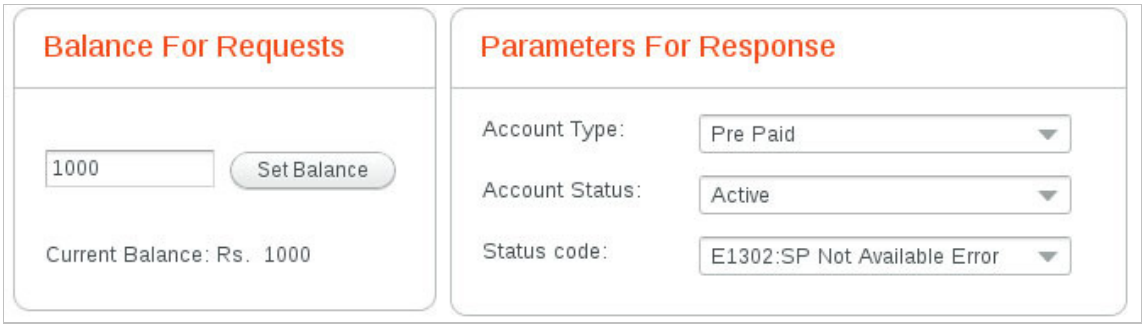

Figure 3.16

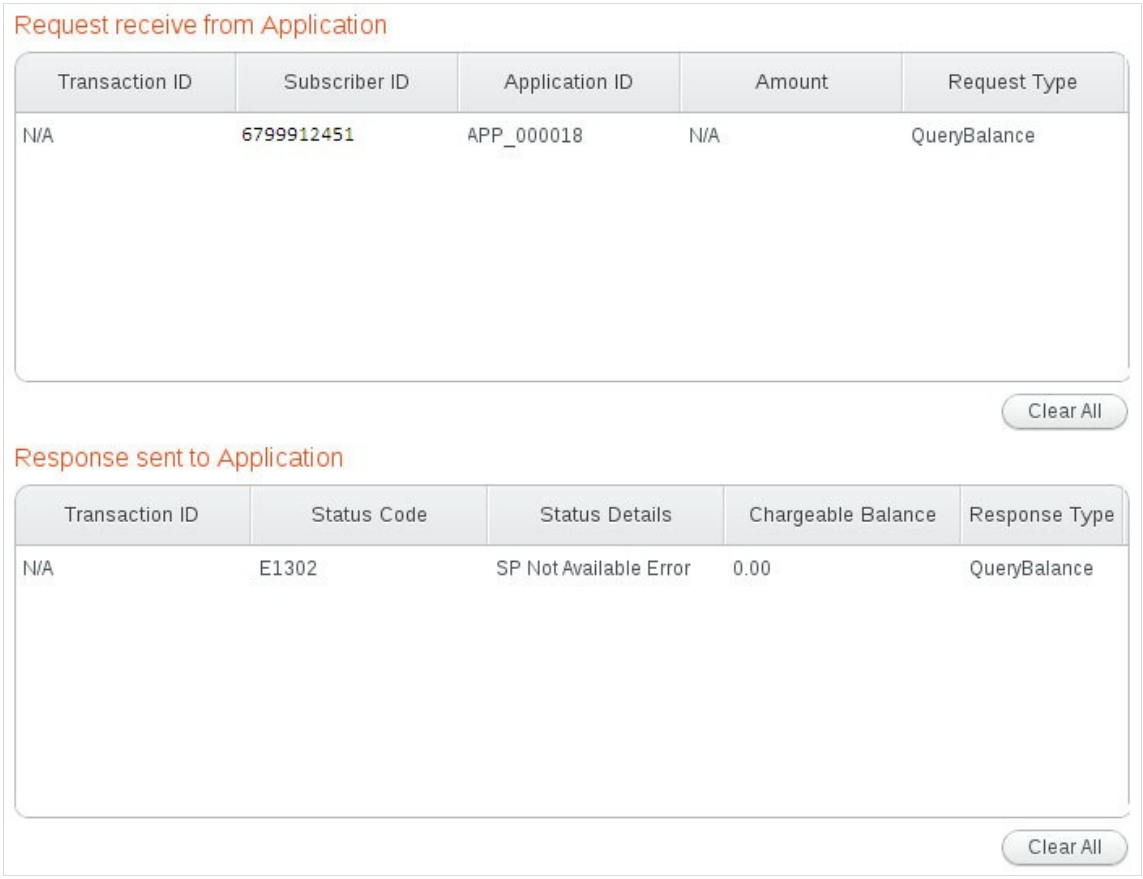

Figure 3.17

Figures 3.16 and 3.17 show a sample response for an error in sending the Query Balance request.

3.1.5.2 Direct Debit

The application has to send the direct debit request to following url

http://localhost:7000/caas/direct/debit

Figure below shows a direct debit transaction of 100 units for an account with a balance of 1000 units.

Figure 3.18

Following is the response.

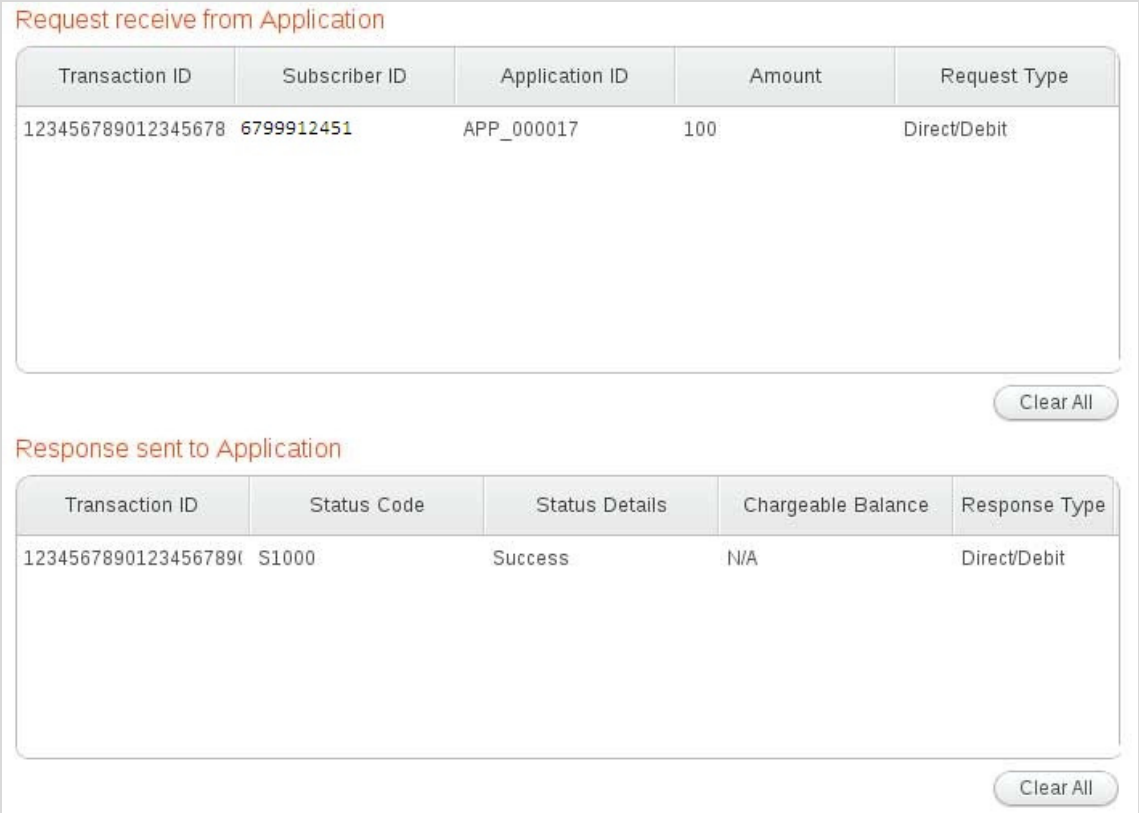

Figure 3.19

The transaction has been done successfully as the account had sufficient balance.

Following screen shows few more successful transactions.

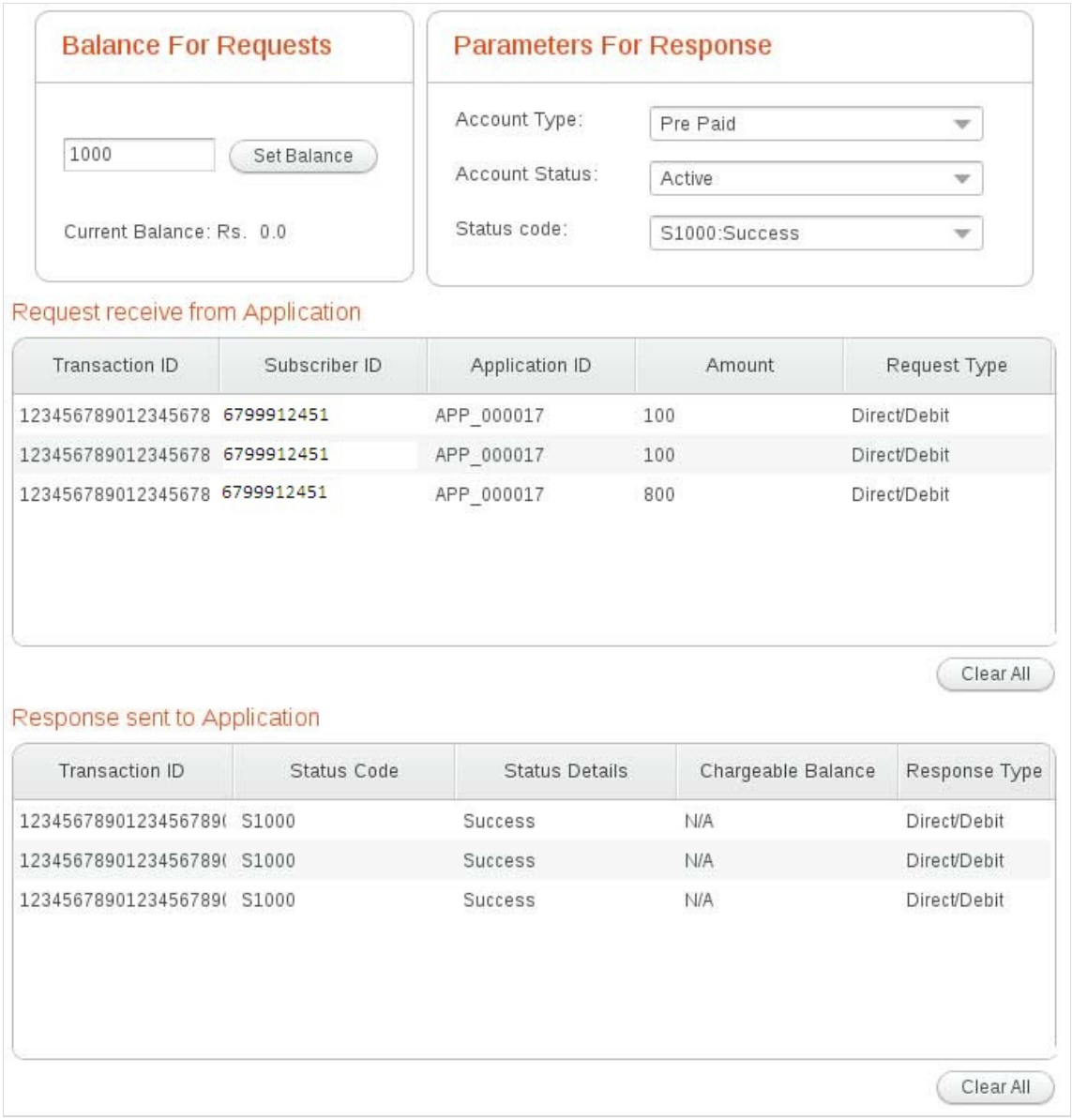

Figure 3.20

Now the account has zero balance. In such scenario, if a transaction is attempted, it will be failed.

Following screen shows a sample of an error scenario with insufficient balance.

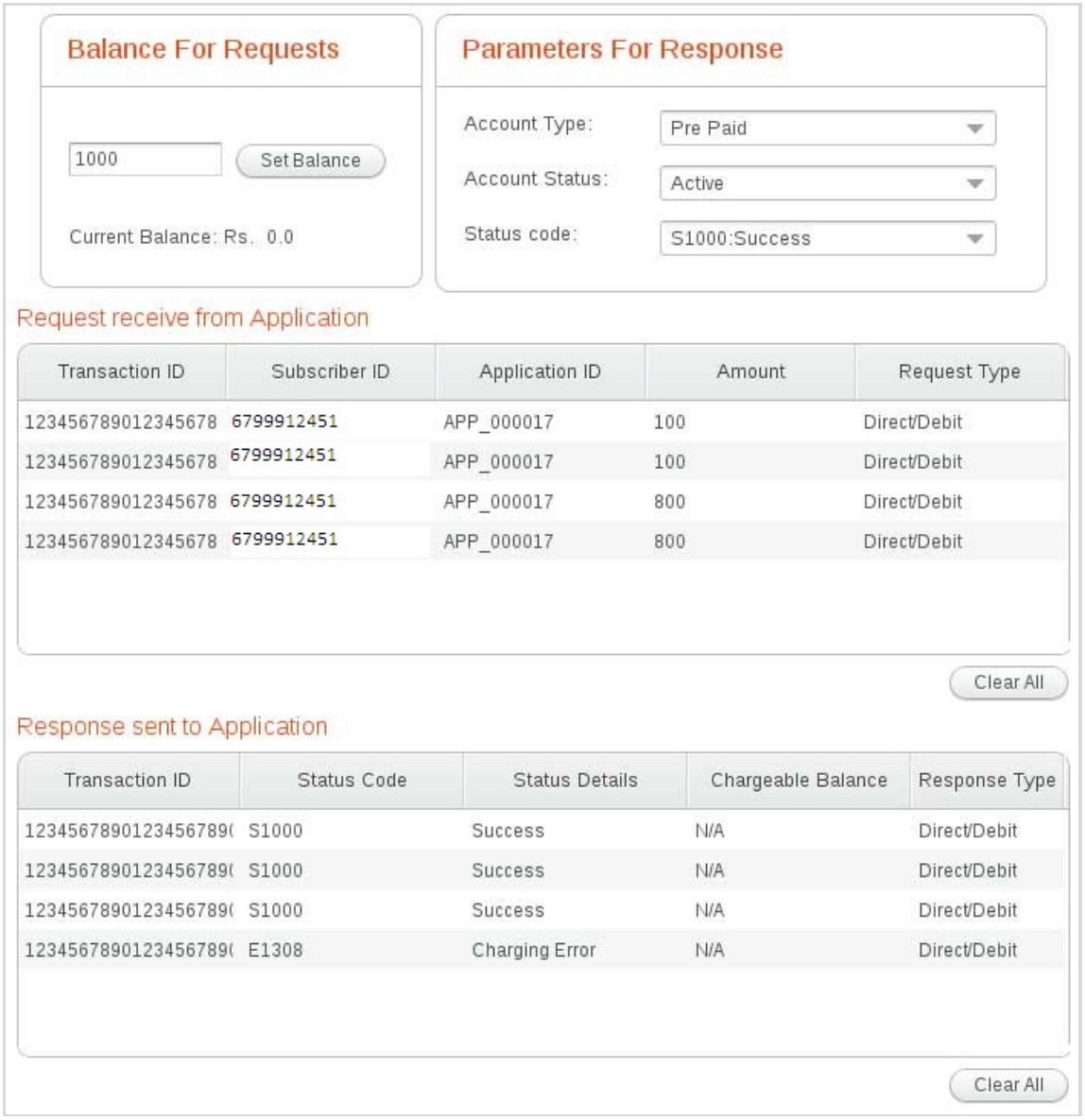

Figure 3.21

In case of an error scenario, refer the following screens for the request and response.

Figure 3.22

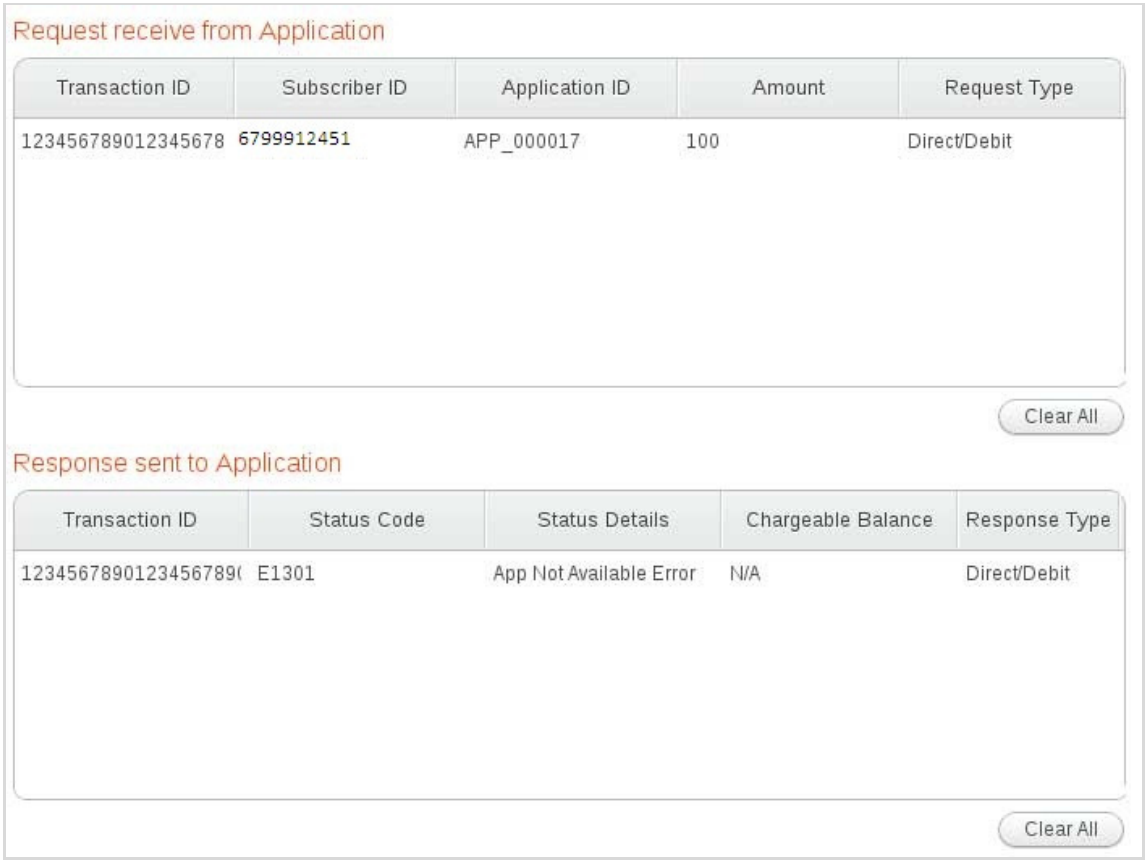

Figure 3.23

3.1.6 Response Error Codes

SDK can respond to a message sent by the client application. A response code will be sent to the destination phone number and the corresponding response shall be identified accordingly.

SDK will send following error codes to the preconfigured phone numbers as given in the list.

For any other number SDK will send success response.

Phone Number	Status Code	Description
9900000	P1001	Partial Success
9900001	E1300	Default Error
9900002	E1301	App Not Available Error
9900003	E1302	SP Not Available Error
9900004	E1303	Invalid Host IP Error
9900005	E1304	App Not Found Error
9900006	E1305	Invalid App ID Error
9900007	E1306	Invalid Routing key Error
9900008	E1307	SP Not Found Error
9900009	E1308	Charging Error
9900010	E1309	NCS Not Allowed Error
9900011	E1310	MO Not Allowed Error
9900012	E1311	MT Not Allowed Error
9900013	E1312	Invalid Request Error
9900014	E1313	Authentication Failed Error
9900015	E1314	NCS Not Available Error
9900016	E1315	App Connection Refused Error
9900017	E1316	MSISDN Not Allowed Error
9900018	E1317	TPS Exceeded Error
9900019	E1318	TPD Exceeded Error

9900020	E1319	AT Message Failed Error
9900021	E1320	SBL Fail Error
9900022	E1321	Sender Not Allowed Error
9900023	E1322	Recipient Not Allowed Error
9900024	E1323	HTTP Request Not Allowed Error
9900025	E1324	Invalid MSISDN Error
9900026	E1325	Insufficient Fund Error
9900027	E1326	Charging Not Allowed Error
9900028	E1327	Charging Operation Not Allowed Error
9900029	E1328	Charging Amount Too High Error
9900030	E1329	Charging Amount Too Low Error
9900031	E1330	Invalid Sender Address Error
9900051	E1350	Subscription Reg Blocked Error
9900052	E1351	Subscription Reg Already Registered Error
9900053	E1352	Subscription Reg SLA Error
9900054	E1353	Subscription Reg Charging Error
9900055	E1354	Subscription Unreg SLA Error
9900056	E1355	Subscription Unreg Blocked Error
9900057	E1356	Subscription Unreg Not Registered Error
9900061	E1360	Internal Error
9900062	E1361	System Error