# Cortex API specification - Prepaid

*Version 2.4.0*

*March 2021*

FIS

www.fisglobal.com

# Statement of Confidentiality

The information contained in or supplied with this document is submitted solely for the purpose of evaluating the products and services of Fidelity National Information Services, Inc., and/or its affiliates and subsidiaries ("FIS"). The information contained in or supplied with this document, in its entirety, is the confidential and proprietary information of FIS, and it may not be copied by or disclosed to any person or entity (other than to the intended recipient), without the prior written consent of FIS. With or without FIS' prior written consent, FIS accepts no liability whatsoever for any consequences arising from the reproduction of the information contained in or supplied with this document, or from its disclosure to any person or entity, including to the intended recipient. FIS additionally accepts no liability for the use of the information contained in, or supplied with this document, by the intended recipient, or by any other person or entity, with or without FIS' express prior consent. The intended recipient shall not use any part of the information contained in, or supplied with this document, in any way to the competitive disadvantage of FIS, and will take all steps designed to assure its compliance with this provision.

This proposal is neither an offer nor intended by FIS, upon acceptance by the intended recipient, or otherwise, to create a binding agreement with FIS. Such an agreement shall be reflected only by a definitive contract, executed by both parties.

# Trademarks

All other trademarks are the property of their owners.

Company, product, and service names used by Fidelity National Information Services, Inc. within, or supplied with this document may be trademarks or service marks of other persons or entities.

# Copyright

# Document Revision History

| | | | | |
|---|---|---|---|---|
| **Name of the document:** Cortex API specification for Prepaid – REST – JSON format | | | | |
| Version 2.4.0 | | Document ID: | | |
| **Document Definition:** Cortex API specification for Prepaid – REST – JSON format | | | | |
| **Author:** R Prabakaran / P Kabilan | | Effective Date: | 19-March-2021 | |
| Maintained by: | | | | |
| **Reviewed by:** Rajendran Suresh, Anbazhagan Chandran | | | | |
| **Approved by:** | | | | |
| **Version Number** | **Revision Description** | **Date** | **Change Author** | **Change Approver** |
| 1.0 | Draft version | 14-Jul-2020 | Kumari Jyoti / R Prabakaran | Rajendran Suresh |
| 2.0 | Clear PAN, SHA256 PAN, and cvv2 added based on request for card linking/activation API, customer details fetch API and Virtual card creation API | 15-Jul-2020 | Prabakaran R | Rajendran Suresh |
| 2.1 | Gift Card ordering, KYC update in customer Update, Verify CVV2 and Channel and Limit update APIs | 11-Aug-2020 | Prabakaran R | Suresh R |
| 2.2 | Card linking / activation service account no. is optional and Card Replacement Non Perso service account no. is optional. | 13-Oct-2020 | Prabakaran R | Suresh R |
| 2.3 | Card Upgrade [Card Product Change] service added, Virtual card creation account no. and account type is optional and document alignment issue fixed | 22-Oct-2020 | Prabakaran R | Suresh R |
| 2.3.1 | Channel Update service name change to **channelupdate** | 20-Nov-2020 | Prabakaran R | Suresh R |
| 2.3.2 | AccountLoad, BalanceAdjustment, Cvv Verification service are modified | 04-Dec-2020 | Prabakaran R | Suresh R |
| 2.3.3 | Card Details Enquiry Modified | 09-Dec-2020 | Prabakaran R | Suresh R |
| 2.3.4 | Account Load, Balance Adjustment amtbill length change.<br>Card Order Standard Perso Service removed | 15-Dec-2020 | Prabakaran R | Suresh R |

| | | | | |
|---|---|---|---|---|
| 2.3.5 | Cvv Verification service mobile number and custcode fields are added in response<br>Card order standard perso service added | 18-01-2021 | Prabakaran R | Suresh R |
| 2.3.6 | Adjust Account Balance, Account Load, Set PIN, Change PIN, Transaction details Enquiry service field decription added. | 20-01-2021 | Prabakaran R | Suresh R |
| 2.3.7 | Gender & marital status is change to optional, gmail keyword rename to email. | 25-01-2021 | Prabakaran R | Suresh R |
| 2.3.8 | Error code message are changed.<br>Authman note added. | 29-01-2021 | Prabakaran R | Suresh R |
| 2.3.9 | Cust code to be made mandatory in Card linking, standard perso APIs | 02-02-2021 | Prabakaran R | Suresh R |
| 2.3.10 | Custcode length reduced 12 to 10 | 10-03-2021 | Prabakaran R | Suresh R |
| 2.4.0 | ClearPan, cvv2, ExpDate response fields are encrypted for security purpose.<br>Using Authman Session key to decrypt. | 19-03-2021 | Prabakaran R | Suresh R |

Document Distribution List

| **Sr. No** | **Associate/ Group Name** | **Organization/ Team** | **Purpose** |
|---|---|---|---|
| 1 | PSS | FIS | Reference |

# Table of Contents

# 1. Introduction

## Purpose & Scope

This document provides the specifications that should be adhered to within the banks via the Cortex web service API. In addition to the I/O specification accompanying text descriptions provide reference to explain the usage of specific data elements related to this specific solution together with any validation that are implemented within the web services in addition to other relevant behavior they exhibit.

**Important note**: This is a technical specification document and does not provides scope of project or SOW.

# 2. Authman

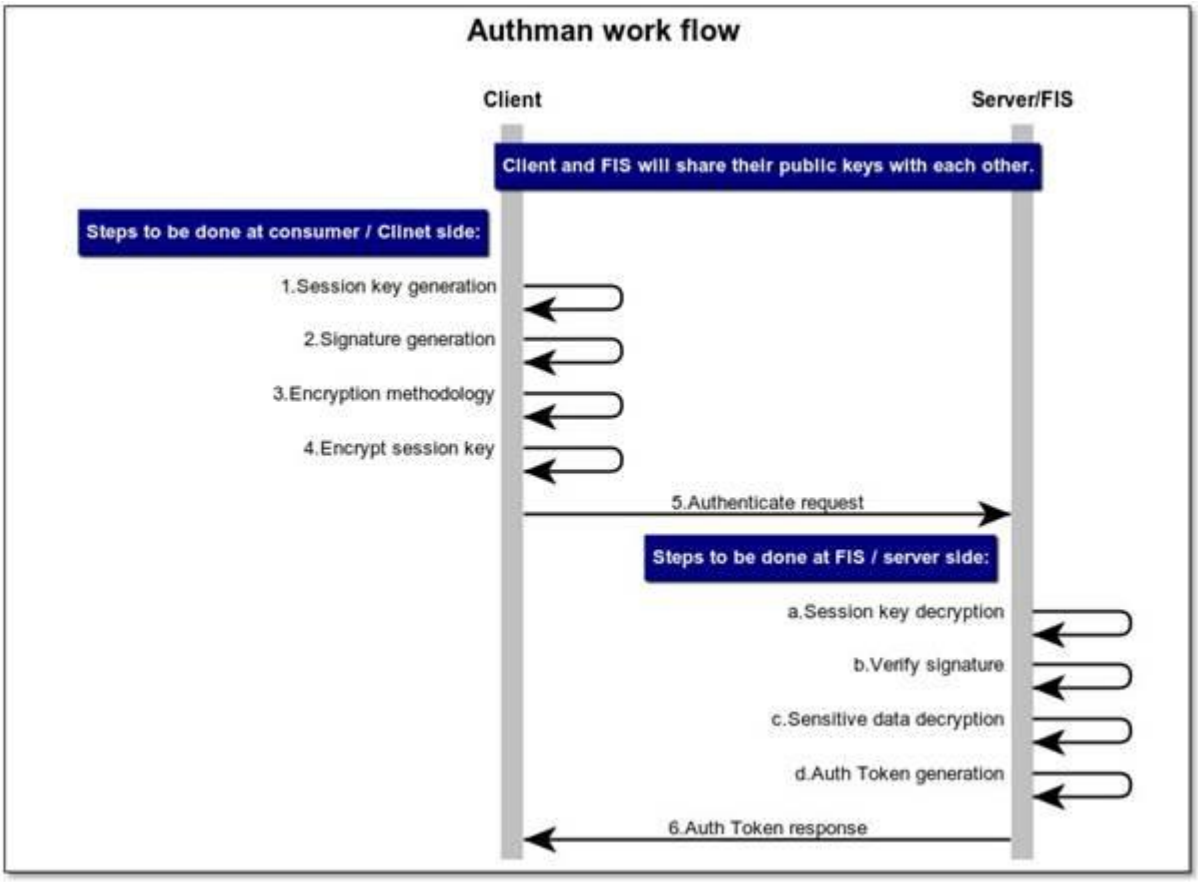
To consume Cortex APIs, first client or calling program or channel should get authenticated. To authenticate, client should call the Authman API.

Below section describes briefly on the Authman version 3.

### Description

This section briefly describes the functionality of Authman version 3.

Below sequence diagram explains the Authman version 3 authentication process flow



Below are the steps to be followed by consumer or Client to Consume Cortex APIs.

**Steps to be done at consumer/client side:**

1. A session key should be generated (Algorithm: AES 256)
2. A signature should be generated with client's private key for the session key (Algorithm: SHA256 with RSA)
3. The institution ID, key alias (user ID) in the request should be encrypted using the session key. (Algorithm: AES/CBC/PKCS5Padding)
   a. IV must be generated randomly for each request
   b. Encrypted value must be converted base64 format
   c. While sending the authenticate request, the parameter should be base64 (encrypted) + "." + base64 (IV)
   d. Java Code snippet is provided in the Annxure 1 for reference

4. Then the session key should be encrypted using the public key of FIS (Algorithm: RSA)
5. The consumer should send the authenticate request to FIS with all these values (in base64 format)
6. Auth Token will be sent in the response by authman API to client or channels which consumes API

**Steps done at FIS side:**

a) When authman receives the request, the session key will be decrypted using the private key of FIS (Algorithm: RSA)

b) The signature sent by the consumer will be verified using the public key of the consumer (Algorithm: SHA256 with RSA)

c) The institution ID, key alias and all other sensitive data in the request will be decrypted using the session key. (Algorithm: AES/CBC/PKCS5Padding)

d) Upon successful authentication (verification of institution ID and user ID), the Auth Token will be generated

### AUTHENTICATE API

This Api is used to validate the signature signed by the client. Once Validation successful we will generate X-Auth-token.

**Note:**

- Generated AuthToken by AuthMan is valid only for 45 minutes.
- Only one auth token is active at a time as soon as new token generated old token should removed/exprire.

**Request URL:**

**URL:** HTTP POST - https://<apihost>:<port>/authman/v3_0/usersec/authenticate

**Request:**

{

```
            “keyAlias”:””,
            “institutionId”:” “
            “appId”:” “,
            “signature”:” “,
            “sessionKey”:” “,
}
```

| Request Parameter | Description |
|---|---|
| keyAlias | keyAlias should be encrypted using the sessionKey<br>Algorithm: AES/CBC/PKCS5Padding |
| institutionId | Institution id will be provided by FIS. It has to be encrypted using Sessionkey (AES)<br>Algorithm: AES/CBC/PKCS5Padding |
| Signature | Signature should be generated using the consumer’s private key.<br>Algorithm: SHA256WithRSA |
| sessionkey | Session key should be encrypted using FIS’s public key. Session key can be any randon 16 or 32 digit alpha numeric<br>Key Algorithm: AES 256<br>Encryption Algorithm: RSA<br>Please refer section 4.4 for the uses of sessionkey |

**Response:**

```
{
      “returnCode”:” “,
       “responseMessage”:” “,
       “errors”: [],
       “access_token”:” “,
       “id_token”: “ “,
       “expires_in”: “ “,
}
```

| Response Code | Description |
|---|---|
| Errors | Error message |
| access_token | Token generated after validation |
| id_token | Unique id of token |
| expires_in | Expiry time of Token in milliseconds |

| | |
|---|---|
| returnCode | 00/01/02/03/04/06 |
| responseMessage | 00 – Success<br>01 – Invalid credentials, please try again<br>02 – Error in Authenticating<br>03 – Error in Decryption<br>04 – User is Locked<br>06 – Invalid Status Code |

**Sample Request:**

```
{
        “keyAlias”:” 0kbOdd5Zc0ccbd7090PtMA==.t7SEZXtjq1vF9JEYUbBXPg==”,
        “ institutionId “:” 0w5xC29wvYvRqQeYBPtajg==.tyayfH/qZw4hfuxAEYlnCA==”,
        “appId”:” “,
        “signature”:”0sjqRg8nkKdxreZam16RgfZavEwEhY62uldUQQn/AjS33appwao6cbM2VlaJfIckl
        2LZHlVoTdANDHaZ4bLuPwhuStrROljeDeiZzyLVvhN69oM/9typX2qWrcH4x6UsoIX+kfp51Yvx
        cYR4AiVyP8rTbINbBX//JMkv7XeTp+6tDfs+Efws0k+YqWqVouwnAnUf6+7HTn096ZsjN9uBW
        u5QcdDtpKOG2A99itfqtQ/O4HuGVwyUJIYB29yUU/Gmub/2mHnBSVu49KqYigLHGR3MpVT
        VMjNxdZELKLjDikdXVwDBe7UZEmS/y/5RriEy68T2bgTsRz6t84QLE6Heew==”,
        “sessionKey”:”mJpiCKZpCJG0dgS5EALL2HLKik+fOSW7MwtgytVQlmg=”
}
```

**Sample Response:**

```
{
          “returnCode”: “00”,
          “responseMessage”: “Success”,
           “errors”: [],
          “access_token”:”eyJhbGciOiJIUzI1NiJ9.eyJhcHBJZCI6IklTVCIsImtleV9pZCI6IjkzZDZmZGFiL
          WFkYTktNGY4MC1hNjIwLWUzNWE2YzgzZjViZSIsInVzZXJJZCI6InRlc3QiLCJpbnN0aXR1dGlv
          bklkIjoiMTIzNDU2IiwicGFzc3dvcmQiOm51bGwsImdyYW50X3R5cGUiOm51bGwsInVzZXJE
          ZXRhaWxzTWFwIjpudWxsLCJqdGkiOm51bGwsImlhdCI6MTUxNzQ3MdcxMzU1MSwiaXNzI
          jpudWxsLCJ1c2VyX2lkIjoidGVzdCIsImlwX2FkZHJlc3MiOm51bGwsImdlb19sb2NhdGlvbiI6b
          nVsbCwibW1pZCI6bnVsbCwibWFwcGVkX3VzZXJfaWQiOm51bGwsImZ1bGxfbmFtZSI6bnV
          sbCwiZW5jcnlwdGlvbl9rZXkiOiJTS1pIV2g3RW9kRGM1QXJ1am4zKytPYlZ2WHFMcUQ3ZFg
          5OG91WDRueURFPSJ9.LjwHRdtYtw3IoutQUKC-Iees54tYYKAPIWgs5s6iB10”,
           “id_token”: “840f334e-bfdd-4eb5-ab75-875f8df48945”,
            “expires_in”: 2700000,
  }
```

## Uses Of Session Key in the API call

- The “SessionKey” is used to encrypt the sensitive values passed in the request.
- Before we use this “sessionKey” to encrypt the sensitive values, this sessionKey has to be decrypted using AES/CBC/PKCS5Padding algorithm.
- This key has to be saved in the session and it has to be regenerated whenever the access_token is regenerated.
- Each request should have added X-Auth-token in header with value of access_token received in authentication response.

**Sample request and response using**:
Header: Content-Type:application/json
X_Auth_Token :

```
“eyJhbGciOiJIUzI1NiJ9.eyJhcHBJZCI6IklTVCIsImtleV9pZCI6IjkzZDZmZGFiLWFkYTktNGY4MC1hNjI
wLWUzNWE2YzgzZjViZSIsInVzZXJJZCI6InRlc3QiLCJpbnN0aXR1dGlvbklkIjoiMTIzNDU2IiwicGFzc3
dvcmQiOm51bGwsImdyYW50X3R5cGUiOm51bGwsInVzZXJEZXRhaWxzTWFwIjpudWxsLCJqdGki
Om51bGwsImlhdCI6MTUxNzQ3MdcxMzU1MSwiaXNzIjpudWxsLCJ1c2VyX2lkIjoidGVzdCIsImlwX
2FkZHJlc3MiOm51bGwsImdlb19sb2NhdGlvbiI6bnVsbCwibW1pZCI6bnVsbCwibWFwcGVkX3VzZ
XJfaWQiOm51bGwsImZ1bGxfbmFtZSI6bnVsbCwiZW5jcnlwdGlvbl9rZXkiOiJTS1pIV2g3RW9kRG
M1QXJ1am4zKytPYlZ2WHFMcUQ3ZFg5OG91WDRueURFPSJ9.LjwHRdtYtw3IoutQUKC-
Iees54tYYKAPIWgs5s6iB10”
```

Body

```
{
   “vpan”:”FISPSS”,
   “email”:”eXC1W8D8jigS2o11zmgMoA==“,
   “mobile_num”:” freYAsPROWW4S0/MviUvFQ==”,
   “Account_num”:”lDdB5K+youWWriMeGBNhCg==”
}
```

**M – Mandatory** **O – Optional** **C-Conditional**

## 3. Card Order – Non-perso (Insta-kit)

Card order non-perso API is used to place card order for insta-kits or non-personalized cards. Only the request will be logged in the system. As part of EOD process, the requested cards will be generated.

**Request Format:**

| S.No. | Field | Min-Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 - 4 | M | AN | Institution Code | |
| 2 | cardOrderNonPersoReq | 1-50 | M | Array | | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | reqMsgId | 18 - 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of cardOrderNonPersoReq Request Object:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | reqId | 18 -18 | M | N | Request id of each request | |
| 2 | actionCode | 1 -1 | M | A | Validation or Execution | V or E |
| 3 | branchCode | 1-8 | M | AN | Branch Code | |
| 4 | cardProduct | 1-4 | M | AN | Card Product | |
| 5 | cardCount | 1-5 | M | N | Card Count | |
| 6 | companyName | 0-50 | O | ANS | This name will be used as emboss name while card creation. | |
| 7 | User | 1-20 | M | ANS | User, who will place card Order | |
| 8 | persoBureauName | 0-20 | O | ANS | | Emboss Vendor Name |
| 9 | userData1 | 0-12 | O | ANS | | User Data 1 ( For Future ) |
| 10 | userData2 | 0-12 | O | ANS | | User Data 2 ( For Future ) |

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | cardOrderNonPersoRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process Code | |
| 4 | reqMsgId | String | Request Message Id | |

|  |  |  |  |  |
|---|---|---|---|---|
| 5 | respCode | String | Response Code |  |
| 6 | respMsg | String | Response message |  |

**Elements of cardRenewRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | cardOrderNonPerResponseList | List of response | Array | Array of Object |

**Array Elements of cardOrderNonPerResponseList Response Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request |  |
| 2 | respCode | String | Response Code |  |
| 3 | respMsg | String | Response Message |  |
| 4 | actionCode | String | Validation or Execution | V or E |
| 5 | branchCode | String | Branch Code |  |
| 6 | cardProduct | String | Card Product |  |
| 7 | cardCount | String | Card Count |  |
| 8 | invoiceNo | String | Random number generated for card order requet | It will be generated only for execution request not for validation. |
| 9 | User | String | User, who will place card Order |  |

**Array Elements of cardOrderNonPerResponseList Response Object for failure response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request |  |
| 2 | respCode | String | Response Code |  |
| 3 | respMsg | String | Response message |  |
| 4 | actionCode | String | Validation or Execution | V or E |

URL : http<s>://<ip>:<port>/app /rest/v1.0/cardordernonperso

**Validation Request** :

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“cardOrderNonPersoReq”:[
        {
        “reqId” : “201908271919300000”,
        “actionCode”:”V”,
        “branchCode” : “0001”,
        “cardProduct” : “MYMC”,
        “cardCount” : “00001”,
        “user” : “user1”
        },
        {
        “reqId” : “201908271919300001”,
        “actionCode”:”V”,
        “branchCode” : “0001”,
        “cardProduct” : “MYMC”,
        “cardCount” : “00001”,
        “user” : “user2”
        }]
}
```

**Validation Response :**

```
{
           “respCode”: “901”,
          “respMsg”: “Request Processed.”,
          “processCode”: 101,
          “instCode”: “CLNT”,
         “reqMsgId” : “201951968545010001”,

   “cardOrderNonPersoRes”: {

          “cardOrderNonPerResponseList”: [
            {
              “reqId”: “201908271919300000”,
              “respCode”: “00”,
              “respMsg”: “Non Perso Card Order Validation Done”,
              “actionCode”: “V”,
              “branchCode”: “0001”,
              “cardProduct”: “MYMC”,
              “cardCount”: “00001”,
              “user”: “user1222222222222222”
            },
            {
              “reqId”: “201908271919300001”,
              “respCode”: “00”,
              “respMsg”: “Non Perso Card Order Validation Done”,
              “actionCode”: “V”,
              “branchCode”: “0001”,
              “cardProduct”: “MYMC”,
              “cardCount”: “00001”,
                “user”: “user3”
            }
          ]
        }
}
```

**Execution REQUEST**

```
        {
        “instCode” : “CLNT”,
        “pFlag” : “”,
        “reqMsgId” : “201951968545010001”,

        “cardOrderNonPersoReq”:[
        {
        “reqId” : “201908271919300000”,
        “actionCode”:”E”,
        “branchCode” : “0001”,
        “cardProduct” : “MYMC”,
        “cardCount” : “00001”,
        “user” : “user1”

        },
        {
        “reqId” : “201908271919300001”,
        “actionCode”:”E”,
        “branchCode” : “0001”,
        “cardProduct” : “MYMC”,
        “cardCount” : “00001”,
        “user” : “user2”
        }]
}
```

**SUCCESS RESPONSE**

```
        {
          “respCode”: “901”,
          “respMsg”: “Request Processed.”,
            “processCode”: 101,
           “instCode”: “CLNT”,
       “reqMsgId” : “201951968545010001”,

  “cardOrderNonPersoRes”: {

           “cardOrderNonPerResponseList”: [
             {
                “reqId”: “201908271919300000”,
                “respCode”: “00”,
                “respMsg”: “Bulkiss record processed Successfully”,
                “actionCode”: “E”,
                “branchCode”: “0001”,
                “cardProduct”: “MYMC”,
                “cardCount”: “00001”,
                “invoiceNo”: “000000000446”,
                “user”: “user1”
              },
              {
                “reqId”: “201908271919300001”,
                “respCode”: “00”,
                “respMsg”: “Bulkiss record processed Successfully”,
                “actionCode”: “E”,
                “branchCode”: “0001”,
                “cardProduct”: “MYMC”,
                “cardCount”: “00001”,
                “invoiceNo”: “000000000447”,
                “user”: “user2”
              }
            ]
          }
}
```

**FAILURE RESPONSE**

```
{
            “respCode”: “901”,
           “respMsg”: “Request Processed.”,
            “processCode”: 101,
            “instCode”: “CLNT”,
           “cardOrderNonPersoRes”: {
            “cardOrderNonPerResponseList”: [
              {
                “reqId”: “201908271919300000”,
                “respCode”: “01”,
                “respMsg”: “Branch [00000001] not found”,
                 “actionCode”:”E”

              },
              {
                “reqId”: “201908271919300001”,
                “respCode”: “01”,
                “respMsg”: “Branch [0002] not found”,
                “actionCode”:”E”
              }
            ]}
}
```

## 4. Card Order (insta-kit) cancel

Requested insta-kit or non-perso card order can be cancelled if card is already not generated.

| S.No. | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institution Code | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | cardOrderCnclRequest | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | Pflag | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of cardOrderCnclRequest Object:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 1 | M | A | Validation or Execution | V or E |
| 2 | invoiceNo | 32 | M | AN | Random number generated for every card order request | |
| 3 | User | 0-20 | O | ANS | User, who will place card order | |
| 4 | reqId | 18 | M | N | Request id of each request | |

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | cardOrderCnclRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of cardOrderCnclRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | cardOrderCancelResp | List of response | Array | Array of Object |

**Array Elements of cardOrderCancelResp Response Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response Message | |
| 4 | actionCode | String | Validation or Execution | V or E |
| 5 | invoiceNo | String | Random number generated for card order requet | |

**Array Elements of cardOrderCancelResp Response Object for failure response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response Message | |
| 4 | actionCode | String | Validation or Execution | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/cardordercancel

**Validation REQUEST**

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“cardOrderCnclRequest”:[
 {
“reqId” : “201908271919300000”,
“actionCode”:”V”,
“invoiceNo” : “000000000454”,
“user” : “user1”
},
{
“reqId” : “201908271919300001”,
“actionCode”:”V”,
“invoiceNo” : “000000000453”,
“user” : “user2”
}
]}
```

**VALIDATION SUCCESS RESPONSE**

```
{
 “respCode”: “901”,
 “respMsg”: “Request Processed.”,
  “processCode”: 102,
  “instCode”: “CLNT”,
  “reqMsgId” : “201951968545010001”,
  “cardOrderCnclRes”: {
  “cardOrderCancelResp”: [
   {
     “reqId”: “201909031247330000”,
     “respCode”: “00”,
     “respMsg”: “Validation done successfully”,
     “actionCode”: “V”,
     “invoiceNo”: “000000000454”       },
      {
         “reqId”: “201908271919300001”,
         “respCode”: “00”,
         “respMsg”: “Validation done successfully”,
         “actionCode”: “V”,
         “invoiceNo”: “000000000453”
       }
    ]
}
```

**VALIDATION FAILURE RESPONSE**

```
{
               “respCode”: “901”,
               “respMsg”: “Request Processed.”,
    “processCode”: 102,
   “instCode”: “CLNT”,
  “reqMsgId” : “201951968545010001”,
   “cardOrderCancelResp”: [
       {
         “reqId”: “201909031251350000”,
         “respCode”: “01”,
         “respMsg”: “Given crdorder record not found”
       },
       {
         “reqId”: “201908271919300001”,
         “respCode”: “01”,
        “respMsg”: “Given crdorder record not found”
        }
    ]
}
```

**EXECUTION REQUEST**

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“cardOrderCnclRequest”:[
{
```

```
“reqId” : “201908271919300000”,
“actionCode”:”E”,
“invoiceNo” : “000000000453”,
“user” : “user1”
},
{
“reqId” : “201908271919300001”,
“actionCode”:”E”,
“invoiceNo” : “000000000454”,
“user” : “user2”
}
]}
```

**EXECUTION SUCCESS RESPONSE**

```
{
          “respCode”: “901”,
          “respMsg”: “Request Processed.”,
  “processCode”: 102,
    “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “cardOrderCnclRes”: {
   “cardOrderCancelResp”: [
     {
        “reqId”: “201908271919300000”,
        “respCode”: “00”,
        “respMsg”: “Record Processed Sucessfully”,
        “actionCode”: “E”,
        “invoiceNo”: “000000000453”
     },
     {
        “reqId”: “201908271919300001”,
        “respCode”: “00”,
        “respMsg”: “Record Processed Sucessfully”,
        “actionCode”: “E”,
        “invoiceNo”: “000000000454”,
     }
   ]
}
```

**EXECUTION FAILURE RESPONSE**

```
{
   “respCode”: “901”,
   “respMsg”: “Request Processed.”,
   “processCode”: 102,
   “instCode”: “CLNT”,
   “reqMsgId” : “201951968545010001”,
   “cardOrderCnclRes”: {
   “cardOrderCancelResp”: [
    {
        “reqId”: “201908271919300000”,
        “respCode”: “01”,
       “respMsg”: “Given crdorder record not found”,
       “actionCode”:”E”
    },
    {
      “reqId”: “201908271919300001”,
      “respCode”: “01”,
      “respMsg”: “Given crdorder record not found”,
        “actionCode”:”E”
    }
  ]
}
```

## 5. Card Order (insta-kit) enquiry

Using card order enqiry API, detail regarding non-perso or insta-kit card order details can be fetched.

**Request Format :**

| S.No. | Field | Min-Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institution Code | |
| 2 | cardOrderEnqReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of cardOrderEnqReq Object:**

| Sno | Field | Min-Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 0-1 | O | A | Execution | |
| 2 | invoiceNo | 0-32 | C | AN | Random number generated for every | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | card order request | |
| 3 | fromDate | 0-8 | C | Date | From Date | YYYYMMDD |
| 4 | toDate | 0-8 | C | Date | To Date | YYYYMMDD |
| 5 | branchCode | 0-8 | C | AN | Branch Code | |
| 6 | reqId | 18 | M | N | Request id of each request | |
| 7 | startRecordNo | 0-5 | M | N | Start Record Number | |
| 8 | endRecordNo | 0-5 | M | N | End Record Number | |

**Note: Card order inquiry request should send invoice number only or branch code with Request date range should be given as input.**

***Possibility Like:***

1. ***Only invoice number***
2. ***Branch code with date range***

*Not allowed: [Don't send]*

1. *Invoice number + date range*
2. *Branch code only*
3. *Date range only*

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | cardOrderEnqRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of cardOrderEnqRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | cardOrderEnqResp | List of response | Array | Array of Object |
| 2 | totalRecord | Overa all total No. of record | String | |

**Array Elements of cardOrderEnqResp object for success Execution response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response Message | |
| 4 | actionCode | String | Execution | E |
| 5 | responseBuff | Array | List of response | Array<br>Note: For Already processed response, this field will not be present. |

**Array Elements of responseBuff Object:**

| | | | | |
|---|---|---|---|---|
| 1 | invoiceNo | String | Request invoice number | |
| 2 | cardCount | String | Card Count | |
| 3 | reqDate | Srtring | Request Date | |
| 4 | startBatch | String | Start Batch | |
| 5 | endBatch | String | End Batch | |
| 6 | cardProduct | String | Card Product | |
| 7 | branchDes | String | Branch Description | |
| 8 | orderStatusDesc | String | Order status description | |
| 9 | fromDate | String | Request From Date | YYYYMMDD format<br>From Date from Request body |
| 10 | toDate | String | Request to Date | YYYYMMDD format<br>To Date from Request Body |
| 11 | branchCode | String | Request branch Code | Branch code from request Body |

**Array Elements of cardOrderEnqResp Response Object for failure response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response Message | |
| 4 | actionCode | String | Execution | E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/cardorderenq

**NOTE**: -**Only 50 records can be enquired at a time.**

Validation Request :

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“pFlag”:”s”,
“cardOrderEnqReq”:[
{
“reqId” : “201908311919300010”,
“actionCode”:”V”,
“fromDate”:”20191016”,
“toDate”:”20191016”,
“branchCode” : “0001”,
“startRecordNo”:”1”,
“endRecordNo”:”10”
}
]}
```

Validation Success Response:

```
{
{
     “respCode”: “901”,
      “respMsg”: “Request Processed.”,
  “processCode”: 103,
  “instCode”: “CLNT”,
  “reqId”: “201908311919300010”,
  “reqMsgId” : “201951968545010001”,
  “cardOrderEnqRes”: {
   “totalRecord”:”200”,
    “cardOrderEnqResp”: [
      {
        “reqId”: “201908311919300010”,
        “respCode”: “00”,
        “respMsg”: “Card Order Enquiry – Validation Done”,
        “actionCode”: “V”,
        “fromDate”: “20191016”,
        “toDate”: “20191016”,
        “branchCode”: “0001”
      }
    ]
  }
}
```

Execution Request :

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“pFlag”:”s”,
“cardOrderEnqReq”:[
{
“reqId” : “201908311919300008”,
“actionCode”:”E”,
“invoiceNo”:”000000000082”
}
]}
```

Execution Success Response :

```
{
          “respCode”: “901”,
          “respMsg”: “Request Processed.”,
  “processCode”: 103,
   “instCode”: “CLNT”,
   “reqId”: “2013088116173090 90”,
  “reqMsgId” : “201951968545010001”,

   “cardOrderEnqRes”: {
     “cardOrderEnqResp”: [
       {
          “reqId”: “2013088116173090 90”,
          “respCode”: “00”,
          “respMsg”: “Card order Enquiry Processed Successfully”,
          “actionCode”: “E”,
          “responseBuff”: [
            {
               “invoiceNo”: “000000001607”,
               “cardCount”: “1”,
               “reqDate”: “20191031”,
               “startBatch”: “0”,
               “endBatch”: “0”,
               “cardProduct”: “MAMC”,
               “branchDes”: “CLNT BRANCH”,
               “orderStatusDesc”: “Card Order Placed, But not yet processed”
            }
          ]
       }
     ]
   }
}
```

Failure Response :

```
{
          “respCode”: “901”,
          “respMsg”: “Request Processed.”,
   “processCode”: 103,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “cardOrderEnqRes”: {
   “cardOrderEnqResp”: [
     {
       “reqId”: “201908271919300000”,
       “respCode”: “01”,
       “respMsg”: “Given crdorder record not found”,
“actionCode”:”E”
     },
     {
       “reqId”: “201908271919300000”,
       “respCode”: “01”,
       “respMsg”: “Given crdorder record not found”,
“actionCode”:”E”
     }
   ]
}
```

## 6. Card status Change

Using card card status change API, specific card’s status can be changed.

**Request Format:**

| Sno | Field | Max Length | Mandatory? | DataType | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institution Code | |
| 2 | cardStatusChangeReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of cardStatusChangeReq Object:**

| S. No | Field | Min-Max Length | Mandatory? | DataType | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 1 | M | A | Validation or Excecution | V or E |
| 2 | vpan | 15 - 19 | M | AN | VPAN | |
| 3 | statusDesc | 4 - 20 | M | A | New card status. It should be in one of the below:<br>LOST<br>STOLEN<br>DAMAGE<br>TEMPBLOCK<br>TEMPUNBLOCK | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | INSTAKITDAMAGE | |
| 4 | reason | 0-256 | O | ANS | Reason for card status change | |
| 5 | reqId | 18 | M | N | Request id of each request | |

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | cardStatusChangeRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of cardStatusChangeRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | cardStatusChangeResponse | List of response | Array | Array of Object |

**Array Elements of cardStatusChangeResponse Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response Message | |
| 4 | actionCode | String | Validation or Execution | V or E |
| 5 | vpan | String | VPAN | vpan |
| 6 | statusDesc | String | Status Description | |

**Array Elements of cardStatusChangeRes Object for failure response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response Message | |
| 4 | actionCode | String | Validation or Execution | V or E |

**Status Details:**

| Status Description | Statcode |
|---|---|
| LOST | 04 |
| STOLEN | 05 |
| DAMAGE | 10 |
| TEMPBLOCK | 06 |
| TEMPUNBLOCK | Card previous status |
| INSTAKITDAMAGE | 21 |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/changecardstatus

**Validation Request :**

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“pFlag”:”s”,
“cardStatusChangeReq”:[
{
“reqId” : “201908311919300011”,
“actionCode”:”V”,
“vpan”:”CLNT00000001060”,
“statusDesc”:”LOST”,
“reason” : “lost”
}
]}
```

Validation Response :

```
{
        “respCode”: “901”,
        “respMsg”: “Request Processed.”,
  “processCode”: 106,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “reqId”: “201908311919300011”,
   “cardStatusChangeRes”: {
     “cardStatusChangeResponse”: [
       {
        “reqId”: “201908311919300011”,
        “respCode”: “00”,
        “respMsg”: “Validation done successfully”,
        “actionCode”: “V”,
        “vpan”: “CLNT00000001060”,
        “statusDesc”: “LOST”
       }
     ]
   }
}
```

Execution Request :

```
{
“instCode” : “CLNT”,
“pFlag”:”s”,
“reqMsgId” : “201951968545010001”,
“cardStatusChangeReq”:[
{
“reqId” : “201908311919300012”,
“actionCode”:”E”,
“vpan”:”CLNT00000001060”,
“statusDesc”:”LOST”,
“reason” : “lost”
}
]}
```

Execution Response :

```
{
            “respCode”: “901”,
            “respMsg”: “Request Processed.”,
“processCode”: 106,
   “instCode”: “CLNT”,
   “reqMsgId” : “201951968545010001”,
   “reqId”: “201908311919300012”,
   “cardStatusChangeRes”: {
      “cardStatusChangeResponse”: [
         {
            “reqId”: “201908311919300012”,
            “respCode”: “00”,
            “respMsg”: “Record Processed Sucessfully”,
            “actionCode”: “E”,
            “vpan”: “CLNT00000001060”,
            “statusDesc”: “LOST”
         }
      ]
   }
}
```

Failure Response :

```
{
            “respCode”: “901”,
            “respMsg”: “Request Processed.”,
 “processCode”: 106,
   “instCode”: “CLNT”,
   “reqMsgId”: “201911071595217885”,
   “reqId”: “201908711919300011”,
   “cardStatusChangeRes”: {
      “cardStatusChangeResponse”: [
         {
            “reqId”: “201908711919300011”,
            “respCode”: “01”,
            “respMsg”: “Status[04] change is not allowed for this card[CLNT00000001060]”,
             “actionCode”:”E”
         }
       ]
    }
}
```

## 7. Customer update

Customer update API is used to update customer’s detail and KYC details like name, address, mobile number, email address etc.

Request Format :

| Sno | Field | Max Length | Mandatory? | DataType | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | custUpdateReq | 1-50 | M | Array | List of req | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of custUpdateReq Object:**

| SNo | Field | Max Length | Mandatory? | DataType | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 1 | M | A | Validation or Execution | V or E |
| 2 | custCode | 0 - 10 | C | AN | Customer Code | |
| 3 | Title | 0 - 4 | O | AN | Title of User | |
| 4 | firstName | 0-50 | O | AN | First name of user | |
| 5 | lastName | 0-50 | O | AN | Last name of user | |
| 6 | dateOfBirth | 0 - 8 | O | Date | Date of Birth | |
| 7 | Gender | 1 - 1 | O | A | Gender | |
| 8 | married | 1 - 1 | O | A | Marital Status | |
| 9 | emailId | 0-64 | O | ANS | Email Id | |
| 10 | addressLine0 | 0-75 | O | ANS | Home Address 0 | |
| 11 | addressLine1 | 0-35 | O | ANS | Home Address 1 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 12 | addressLine2 | 0-35 | O | ANS | Home Address 2 | |
| 13 | addressLine3 | 0-35 | O | ANS | Home Address 3 | |
| 14 | homeCity | 0-40 | O | ANS | Home City | |
| 15 | homeCountry | 0-20 | O | AN | Home State/Province | |
| 16 | homeCountryCode | 0-3 | O | N | Home Country Code | |
| 17 | homeTel | 0-20 | O | N | Home Telephone | |
| 18 | homePostCode | 0-10 | O | AN | Home Postal | |
| 19 | Pobox | 0-8 | O | AN | PO Box Number | |
| 20 | workAddRLine1 | 0-30 | O | AN | Work address 1 | |
| 21 | workAddRLine2 | 0-30 | O | AN | Work address 2 | |
| 22 | workAddRLine3 | 0-30 | O | AN | Work address 3 | |
| 23 | workCity | 0-40 | O | AN | Work city | |
| 24 | workCountry | 0-20 | O | AN | Work State/Province | |
| 25 | workCountyCode | 0-3 | O | N | Work Country Code | |
| 26 | workTel | 0-20 | O | N | Work Telephone | |
| 27 | workPostCode | 0-10 | O | AN | Work postal | |
| 28 | idNumber | 0-12 | O | AN | Id Number | |
| 29 | nationalId | 0-22 | O | AN | National Id Number | |
| 30 | mobNo | 0-12 | O | N | Mobile Number | |
| 31 | reqId | 18 | M | N | Id of each request | |
| 32 | accNo | 1-28 | C | AN | Account Number | |
| 33 | kycTypeId | 1 | O | N | Customer KYC:<br>1: NOKYC<br>2: LOW-KYC<br>3: FULL-KYC | |

**Note : ‘Either Account Number or Customer Code is required to update customer details.**

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | custUpdateRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |

|   |   |   |   |   |
|---|---|---|---|---|
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of custUpdateRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | custUpdateResponse | List of response | Array | Array of Object |

**Array Elements of custUpdateResponse Response Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | actionCode | String | Validation or Execution | |
| 5 | custCode | String | Cust Code | If CustCode is provided in request,same will be in response |
| 6 | marriedStatus | String | Marital Status | |
| 7 | emailId | String | Email Id | |
| 8 | accountNo | String | Account Number | If AccountNo is provided in request,same will be in response |

**Array Elements of custUpdateResponse Response Object for failure response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| **1** | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | actionCode | String | Validation or Execution | |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/updatecustprofile

Validation Request:

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“custUpdateReq”:[
{
```

```
“reqId” : “201908271919300000”,
“actionCode”:”V”,
“custCode” : “000000100106”,
“title” :”Ms”,
“firstName”:”jabc”,
“lastName” : “def”,
“dateOfBirth” :”19940507”,
“gender”:”F”,
“married” : “S”,
“emailId” :”test@xyz.com”,
“addressLine0”:”ADDR0”,
“addressLine1” : “ADDR1”,
“addressLine2” :”ADDR2”,
“addressLine3”:”ADDR3”,
“homeCity” : “Chennai”,
“homeCountry” :”TN”,
“homeCountryCode” : “356”,
“homeTel” :”987654321212”,
“homePostCode” : “675432”,
“pobox” :”89765432”,
“workAddRLine1” : “WADDR1”,
“workAddRLine2” :”WADDR2”,
“workAddRLine3” : “WADDR3”,
“workCity” :”Chennai”,
“workCountry” :”TNB”,
“workCountryCode” : “356”,
“workTel” :”789765432134”,
“workPostCode” :”674312”,
“idNumber” : “12121”,
“nationalId” :”90909090”,
“mobNo” :”90909090909”
},
{
“reqId” : “201908271919300001”,
“actionCode”:”V”,
“custCode” : “000000100106”,
“title” :”Ms”,
“firstName”:”jabc”,
“lastName” : “def”,
“dateOfBirth” :”19940507”,
“gender”:”F”,
“married” : “S”,
“emailId” :”test@xyz.com”,
“addressLine0”:”ADDR0”,
“addressLine1” : “ADDR1”,
“addressLine2” :”ADDR2”,
“addressLine3”:”ADDR3”,
“homeCity” : “Chennai”,
“homeCountry” :”TN”,
“homeCountryCode” : “356”,
“homeTel” :”987654321212”,
“homePostCode” : “675432”,
“pobox” :”89765432”,
“workAddRLine1” : “WADDR1”,
“workAddRLine2” :”WADDR2”,
“workAddRLine3” : “WADDR3”,
“workCity” :”Chennai”,
“workCountry” :”TNB”,
“workCountryCode” : “356”,
“workTel” :”789765432134”,
“workPostCode” :”674312”,
“idNumber” : “12121”,
“nationalId” :”90909090”,
“mobNo” :”90909090909”
}
]}
```

Validation Response:

```
{
{
“respCode”: “901”,
“respMsg”: “Request Processed.”,
 “processCode”: 109,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “custUpdateRes”: {
 “custUpdateResponse”: [
       {
          “reqId”: “201908271919300000”,
          “respCode”: “00”,
          “respMsg”: “Customer Validation Done”,
          “actionCode”: “V”,
          “custCode”: “000000100106”,
          “marriedStatus”: “S”,
          “emaild”: “test@xyz.com”
       },
       {
          “reqId”: “201908271919300001”,
          “respCode”: “00”,
          “respMsg”: “Customer Validation Done”,
          “actionCode”: “V”,
          “custCode”: “000000100106”,
          “marriedStatus”: “S”,
          “emailId”: “test@xyz.com”
       }
   ]
}
```

Execution Request:

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“custUpdateReq”:[
{
“reqId” : “201908271919300000”,
“actionCode”:”E”,
“custCode” : “000000100106”,
“title” :”Ms”,
“firstName”:”jabc”,
“lastName” : “def”,
“dateOfBirth” :”19940507”,
“gender”:”F”,
“married” : “S”,
“emailId” :”test@xyz.com”,
“addressLine0”:”ADDR0”,
“addressLine1” : “ADDR1”,
“addressLine2” :”ADDR2”,
“addressLine3”:”ADDR3”,
“homeCity” : “Chennai”,
“homeCountry” :”TN”,
“homeCtry” : “356”,
“homeTel” :”987654321212”,
“homePostCode” : “675432”,
“pobox” :”89765432”,
“workAddRLine1” : “WADDR1”,
“workAddRLine2” :”WADDR2”,
“workAddRLine3” : “WADDR3”,
“workCity” :”Chennai”,
“workCountry” :”TNB”,
“workCtry” : “356”,
“workTel” :”789765432134”,
“workPostCode” :”674312”,
“idNumber” : “12121”,
“nationalId” :”90909090”,
“mobNo” :”90909090909”
},
{
“reqId” : “201908271919300001”,
“actionCode”:”E”,
“custCode” : “000000100106”,
“title” :”Ms”,
“firstName”:”jabc”,
“lastName” : “def”,
“dateOfBirth” :”19940507”,
“gender”:”F”,
“married” : “S”,
“emailId” :”test@xyz.com”,
“addressLine0”:”ADDR0”,
“addressLine1” : “ADDR1”,
“addressLine2” :”ADDR2”,
“addressLine3”:”ADDR3”,
“homeCity” : “Chennai”,
“homeCountry” :”TN”,
“homeCtry” : “356”,
“homeTel” :”987654321212”,
“homePostCode” : “675432”,
“pobox” :”89765432”,
“workAddRLine1” : “WADDR1”,
“workAddRLine2” :”WADDR2”,
“workAddRLine3” : “WADDR3”,
“workCity” :”Chennai”,
“workCountry” :”TNB”,
“workCtry” : “356”,
“workTel” :”789765432134”,
“workPostCode” :”674312”,
“idNumber” : “12121”,
“nationalId” :”90909090”,
“mobNo” :”90909090909”
}
]}
```

Execution Response:

```
{
“respCode”: “901”,
“respMsg”: “Request Processed.”,
“processCode”: 109,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “custUpdateRes”: {
   “custUpdateResponse”: [
      {
         “reqId”: “201908271919300000”,
         “respCode”: “00”,
         “respMsg”: “Customer Updated”,
         “actionCode”: “E”,
         “custCode”: “000000100106”,
         “marriedStatus”: “S”,
         “emailId”: “test@xyz.com”
      },
      {
         “reqId”: “201908271919300001”,
         “respCode”: “00”,
         “respMsg”: “Customer Updated”,
         “actionCode”: “E”,
         “custCode”: “000000100106”,
         “marriedStatus”: “S”,
         “emailId”: “test@xyz.com”
      }
   ]
}
```

Failure Response:

```
{
{
“respCode”: “901”,
“respMsg”: “Request Processed.”,
  “processCode”: 109,
  “instCode”: “CLNT”,
 “reqMsgId” : “201951968545010001”,
  “custUpdateRes”: {
  “custUpdateResponse”: [
    {
      “reqId”: “201908271919300000”,
      “respCode”: “01”,
      “respMsg”: “Error in parse record”,
    “actionCode”:”E”

    },
    {
      “reqId”: “201908271919300001”,
      “respCode”: “01”,
      “respMsg”: “Error in parse record”,
     “actionCode”:”E”

    }
  ]
}
```

## 8. Card Replacement (using Non-perso or Insta-kit)

Card replacement API used to replace a cancelled (hot) card using insta-kit or non-perso card.

| SNo | Field | Max Length | Mandatory | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institution Code | |
| 2 | cardReplacementNonPersoReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of cardReplacementNonPersoReq Object:**

| S.No | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 1 - 1 | M | A | Validation or Execution | V or E |
| 2 | oldVpan | 15 - 19 | M | AN | Old Vpan Number | |
| 3 | newVpan | 15 - 19 | M | AN | New Vpan Number | |
| 4 | accountNo | 0 - 28 | O | AN | Account Number | |
| 5 | reqId | 18 - 18 | M | N | Id of each request | |

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | cardReplaceNonPersoRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of cardReplaceNonPersoRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | cardReplaceNonPersoResponse | List of response | Array | Array of Object |

**Array Elements of cardReplaceNonPersoResponse Response Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| **1** | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | actionCode | String | Validation or Execution | |
| 5 | oldVpan | String | Old vpan | |
| 6 | newVpan | String | New vpan | |
| 7 | accountNo | String | Account number | |

**Array Elements of cardReplaceNonPersoResponse Response Object for failure response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | actionCode | String | Validation or Execution | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/cardreplacenonperso

Validation Request :

{

```
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“cardReplacementNonPersoReq”:[
{
 “reqId” : “201908271919300000”,
 “actionCode”:”V”,
  “oldVpan” :”MYMC000000000333”,
   “newVpan” :”MYMC000000000762”,

 “accountNo” :”000000000003 “
}
]}
```

Validation Response :

```
{
  “respCode”: “901”,
“respMsg”: “Request Processed.”,

   “processCode”: 110,
    “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
    “cardReplaceNonPersoRes”: {
    “cardReplaceNonPersoResponse”: [
        {
            “reqId”: “201908271919300000”,
            “respCode”: “00”,
            “respMsg”: “Card Replacement Validation done Successfully”,
            “actionCode”: “V”,
            “oldVpan”: “MYMC000000000333”,
            “newVpan”: “MYMC000000000762”,
            “accountNo”: “000000000003 “
        }
    ]
}
```

Execution Request :

```
{
“respCode”: “00”,
 “respMsg”: “Request Processed Successfully.”,
“processCode”: 110,
    “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
    “cardReplaceNonPersoRes”: {
“cardReplacementNonPersoReq”:[
{
 “reqId” : “201908271919300000”,
 “actionCode”:”E”,
  “oldVpan” :”MYMC000000000333”,
   “newVpan” :”MYMC000000000762”,
 “accountNo” :”000000000003 “
}
]}
```

Execution Response :

```
{
“respCode”: “901”,
“respMsg”: “Request Processed.”,
“processCode”: 110,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “cardReplaceNonPersoRes”: {
  “cardReplaceNonPersoResponse”: [
      {
         “reqId”: “201908271919300000”,
         “respCode”: “00”,
         “respMsg”: “Card Replacement Done Successfully”,
         “actionCode”: “E”,
         “oldVpan”: “MYMC000000000333”,
         “newVpan”: “MYMC000000000762”,
         “accountNo”: “000000000003 “
      }
   ]
}
```

Failure Response :

```
{
“respCode”: “901”,
“respMsg”: “Request Processed.”,
“processCode”: 110,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “cardReplaceNonPersoRes”: {
   “cardReplaceNonPersoResponse”: [
      {
         “reqId”: “201908271919300000”,
         “respCode”: “01”,
         “respMsg”: “Already Replaced Card: vpan: [MYMC000000000077]”
      },
      {
         “reqId”: “201908271919300001”,
         “respCode”: “01”,
         “respMsg”: “Can’t find vpan:[MYMC000000000078]”
      }
   ]
}
```

Failure Res :

```
{
“respCode”: “901”,
“respMsg”: “Request Processed.”,
“processCode”: 110,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “cardReplaceNonPersoRes”: {
“cardReplaceNonPersoResponse”: [
      {
         “reqId”: “201908271919300000”,
         “respCode”: “01”,
         “respMsg”: “Given New Card Not in Valid Status”
      }
   ]
}
```

## 9. Card Order / Card Issue – Standard-perso

Card order standard perso API used to issua a card to a customer.

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | cardOrderStandardPersoReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of cardOrderStndrdPersoReq Object:**

| S. No | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 1 | M | A | Validation or Execution | V or E |
| 2 | branchCode | 1-8 | M | AN | Branch Code | |
| 3 | cardProduct | 1-4 | M | AN | Card Product | MAMX(for amex card) |
| 4 | custCode | 1-10 | M | AN | Customer Code | |
| 5 | custType | 0-5 | O | N | Customer Type | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | title | 0-4 | O | ANS | Title of customer | |
| 7 | firstName | 2-50 | M | ANS | Customer’s first name | |
| 8 | lastName | 1-50 | M | ANS | Customer’s last name | |
| 9 | birthDate | 0-8 | O | Date | Birth date | |
| 10 | gender | 1 - 1 | M | A | Gender | |
| 11 | marriedStatus | 1 - 1 | M | A | Marital Status | |
| 12 | emailId | 0-64 | O | AN | Email Id | |
| 13 | profession | 0-5 | O | N | Profession | |
| 14 | homeAddr1 | 0-35 | O | ANS | Home Address 1 | |
| 15 | homeAddr2 | 0-35 | O | ANS | Home Address 2 | |
| 16 | homeAddr3 | 0-35 | O | ANS | Home Address 3 | |
| 17 | homeCity | 0-40 | O | ANS | Home City | |
| 18 | homeCounty | 0-20 | O | AN | Home State/Province | |
| 19 | homeCountry | 0-3 | O | N | Home Country Code | |
| 20 | homeTel | 0-20 | O | N | Home Telephone | |
| 21 | homePostalCode | 0-10 | O | AN | Home postal code | |
| 22 | poBox | 0-8 | O | AN | Po Box | |
| 23 | workAddr1 | 0-30 | O | AN | Work address 1 | |
| 24 | workAddr2 | 0-30 | O | AN | Work address 2 | |
| 25 | workAddr3 | 0-30 | O | AN | Work address 3 | |
| 26 | workCity | 0-40 | O | AN | Work city | |
| 27 | workCounty | 0-20 | O | AN | Work State/Province | |
| 28 | workCountry | 0-3 | O | N | Work Country Code | |
| 29 | workTel | 0-20 | O | N | Work telephone | |
| 30 | workPostalCode | 0-10 | O | AN | Work postal code | |
| 31 | identificationNumber | 0-12 | O | AN | Id number | |
| 32 | nationalId | 0-22 | O | AN | National Identity | |
| 33 | mobileNumber | 0-20 | O | N | Mobile number | |
| 34 | fax | 0-20 | O | AN | Fax | |
| 35 | countryName | 0-25 | O | AN | Country Name | |
| 36 | nationality | 0-50 | O | AN | Nationality | |
| 37 | suitNumber | 0-8 | O | AN | Suit Number | |
| 38 | accountType | 1 - 2 | M | AN | Account Type | |
| 39 | accountNo | 1 - 28 | M | AN | Account Number | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 40 | accountStatus | 0-2 | O | N | Account status | |
| 41 | embossName | 0-25 | O | ANS | Emboss name | |
| 42 | reqId | 1-8 | M | N | Id of each request | |

**Response Format:**

| S.No. | Field | Description | Comments |
|---|---|---|---|
| 1 | instCode | Institution Code | |
| 2 | cardOrderStdPersoRes | JSON Object | JSON Object |
| 3 | processCode | Number | Process id of each request |
| 4 | reqMsgId | String | Request Message Id |
| 5 | respCode | String | Response Code |
| 6 | respMsg | String | Response message |

**Elements of cardOrderStdPersoRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | cardOrderStdPerResponse | List of response | Array | Array of Object |

**Array Elements of cardOrderStdPerResponse Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| **1** | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | clrPan | String | Clear Pan value encrypted by Authman session key | It will be generated only for execution request not for validation.<br>Clear Pan value encrypted by Authman session key |

| | | | | |
|---|---|---|---|---|
| 5 | vpan | String | Vpan | It will be generated only for execution request not for validation. |
| 6 | panDispaly | String | Pan Display | It will be generated only for execution request not for validation. |
| 7 | encryptedPan | String | Ecncryted Pan | It will be generated only for execution request not for validation. |
| 8 | iid | String | Iid | It will be generated only for execution request not for validation. |
| 9 | expDate | String | Expiry Date | It will be generated only for execution request not for validation. |
| 10 | actionCode | String | Action code | |
| 11 | branchCode | String | Branch Code | |
| 12 | cardProduct | String | Card Product | |
| 13 | firstName | String | First Name | |
| 14 | lastName | String | Last Name | |
| 15 | gender | String | Gender | |
| 16 | marriedStatus | String | Marital Status | |
| 17 | accountType | String | Account Type | |
| 18 | accountNo | String | Account Number | |
| 19 | embossName | String | Emboss Name | |

**Array Elements of cardOrderStdPerResponse Object for failure response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | actionCode | String | Validation or Execution | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/cardorderstanperso

Validation Request :

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“cardOrderStdPersoReq”:[
{
 “reqId” : “201908271919300000”,
 “actionCode”:”V”,
  “branchCode” :”0001”,
 “cardProduct” :”MYMC”,
 “firstName” : “CUSTOMER”,
 “lastName” : “CLNT”,
 “gender” : “F”,
 “marriedStatus” : “S”,
 “accountType” : “12”,
 “accountNo” : “0000000012020301”,
 “embossName” : “CORTEX”

},
{
“reqId” : “201908271919300001”,
 “actionCode”:”V”,
 “branchCode” :”0001”,
 “cardProduct” :”MYMC”,
 “firstName” : “CUSTOMER”,
 “lastName” : “CLNT”,
 “gender” : “F”,
 “marriedStatus” : “S”,
 “accountType” : “12”,
 “accountNo” : “0000000012020302”,
 “embossName” : “CORTEX”
}
]}
```

Validation Response :

```
{
{
   “respCode”: “901”,
 “respMsg”: “Request Processed.”,
  “processCode”: 113,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “cardOrderStdPersoRes”: {
  “cardOrderStdPerResponse”: [
        {
           “reqId”: “201908271919300000”,
           “respCode”: “00”,
           “respMsg”: “Validation Done – Given customer name and account linked customer name
are matched”,
           “actionCode”: “V”,
           “branchCode”: “0001”,
           “cardProduct”: “MYMC”,
           “firstName”: “CUSTOMER”,
           “lastName”: “CLNT”,
           “gender”: “F”,
           “marriedStatus”: “S”,
           “accountType”: “12”,
           “accountNo”: “0000000012020301”,
           “embossName”: “CORTEX”
        },
        {
           “reqId”: “201908271919300001”,
           “respCode”: “00”,
           “respMsg”: “Validation Done – Given customer name and account linked customer name
are matched”,
           “actionCode”: “V”,
           “branchCode”: “0001”,
           “cardProduct”: “MYMC”,
           “firstName”: “CUSTOMER”,
           “lastName”: “CLNT”,
           “gender”: “F”,
           “marriedStatus”: “S”,
           “accountType”: “12”,
           “accountNo”: “0000000012020302”,
           “embossName”: “CORTEX”
        }
    ]
}
```

Execution Request :

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“cardOrderStdPersoReq”:[
{
 “reqId” : “201908271919300000”,
 “actionCode”:”E”,
  “branchCode” :”0001”,
 “cardProduct” :”MYMC”,
 “firstName” : “CUSTOMER”,
 “lastName” : “CLNT”,
 “gender” : “F”,
 “marriedStatus” : “S”,
 “accountType” : “12”,
 “accountNo” : “0000000012020301”,
 “embossName” : “CORTEX”

},
{
“reqId” : “201908271919300001”,
 “actionCode”:”E”,
 “branchCode” :”0001”,
 “cardProduct” :”MYMC”,
 “firstName” : “CUSTOMER”,
 “lastName” : “CLNT”,
 “gender” : “F”,
 “marriedStatus” : “S”,
 “accountType” : “12”,
 “accountNo” : “0000000012020302”,
 “embossName” : “CORTEX”
}
```

]}

Execution Response :

```
{
{
   “respCode”: “901”,
 “respMsg”: “Request Processed.”,
   “processCode”: 113,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “cardOrderStdPersoRes”: {
   “cardOrderStdPerResponse”: [
       {
          “reqId”: “201908271919300000”,
          “respCode”: “00”,
          “respMsg”: “Card Order Standard Perso Processed Successfully”,
          “clrPan”: “ “,
          “vpan”: “MYMC000000000895”,
          “panDispaly”: “223005_6125”,
          “encryptedPan”: “D56087C92BA73910829”,
          “iid”: “223005”,
          “actionCode”: “E”,
          “branchCode”: “0001”,
          “cardProduct”: “MYMC”,
          “firstName”: “CUSTOMER”,
          “lastName”: “CLNT”,
          “gender”: “F”,
          “marriedStatus”: “S”,
          “accountType”: “12”,
          “accountNo”: “0000000012020301”,
          “embossName”: “CORTEX”
       },
       {
          “reqId”: “201908271919300001”,
          “respCode”: “00”,
          “respMsg”: “Card Order Standard Perso Processed Successfully”,
          “clrPan”: “ “,
          “vpan”: “MYMC000000000903”,
          “panDispaly”: “223005_9448”,
          “encryptedPan”: “D8653E6F820D2E9C572”,
          “iid”: “223005”,
          “actionCode”: “E”,
          “branchCode”: “0001”,
          “cardProduct”: “MYMC”,
          “firstName”: “CUSTOMER”,
          “lastName”: “CLNT”,
          “gender”: “F”,
          “marriedStatus”: “S”,
          “accountType”: “12”,
          “accountNo”: “0000000012020302”,
          “embossName”: “CORTEX”
```

```
      }
   ]
}
```

Failure Response :

```
{
{
   “respCode”: “901”,
 “respMsg”: “Request Processed.”,
“processCode”: 113,
   “instCode”: “CLNT”,
“reqMsgId” : “201951968545010001”,
   “cardOrderStdPersoRes”: {
   “cardOrderStdPerResponse”: [
      {
         “reqId”: “2019082719193000000”,
         “respCode”: “01”,
         “respMsg”: “Error: Customer Firstname and Lastname mismatch”,
        “actionCode”:”E”

      },
      {
         “reqId”: “2019082719193000001”,
         “respCode”: “01”,
         “respMsg”: “Error: Customer Firstname and Lastname mismatch”,
         “actionCode”:”E”

      }
   ]
          }
```

## 10. Card Linking/Activation

Using card link API, insta-kit or non-perso card can be linked to account and customer (after creating customer and account).

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | cardLinkReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 - 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of cardLinkReq Object:**

Note: Either pan or vpan or custCode should mandatory

| S.No | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | reqId | 18 | M | N | Id of each request | |
| 2 | actionCode | 1 | M | A | Validation or Execution | |
| 3 | vpan | 15 - 19 | CO | AN | Unique identifier of card | Either pan or vpan should mandatory |
| 4 | pan | 17-19 | CO | N | Full length of Pan | Either pan or vpan should mandatory |
| 5 | custCode | 1-10 | M | AN | Customer Code | |
| 6 | custType | 0-5 | O | N | Customer Type | |
| 7 | title | 0-4 | O | ANS | Title of customer | |
| 8 | firstName | 1-50 | M | ANS | Customer’s first name | |
| 9 | lastName | 1-50 | M | ANS | Customer’s last name | |
| 10 | birthDate | 0-8 | O | Date | Birth date | |
| 11 | gender | 1 | M | A | Gender | |
| 12 | marriedStatus | 1 | M | A | Marital Status | |
| 13 | emailId | 0-64 | O | AN | Email Id | |
| 14 | profession | 0-5 | O | N | Profession | |
| 15 | homeAddr1 | 0-35 | C | ANS | Home Address 1 | Work address or Home address is mandatory |
| 16 | homeAddr2 | 0-35 | C | ANS | Home Address 2 | Work address or Home address is mandatory |
| 17 | homeAddr3 | 0-35 | C | ANS | Home Address 3 | Work address or Home address is mandatory |
| 18 | homeCity | 0-40 | C | ANS | Home City | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 19 | homeCounty | 0-20 | C | AN | Home State/Province | |
| 20 | homeCountry | 0-3 | C | N | Home Country Code | |
| 21 | homeTel | 0-20 | C | N | Home Telephone | |
| 22 | homePostalCode | 0-10 | C | AN | Home postal code | |
| 23 | poBox | 0-8 | O | AN | Po Box | |
| 24 | workAddr1 | 0-30 | C | AN | Work address 1 | Work address or Home address is mandatory |
| 25 | workAddr2 | 0-30 | C | AN | Work address 2 | Work address or Home address is mandatory |
| 26 | workAddr3 | 0-30 | C | AN | Work address 3 | Work address or Home address is mandatory |
| 27 | workCity | 0-40 | C | AN | Work city | |
| 28 | workCounty | 0-20 | C | AN | Work State/Province | |
| 29 | workCountry | 0-3 | C | N | Work Country Code | |
| 30 | workTel | 0-20 | C | N | Work telephone | |
| 31 | workPostalCode | 0-10 | C | AN | Work postal code | |
| 32 | identificationNumber | 0-12 | O | AN | Id number | |
| 33 | nationalId | 0-22 | O | AN | National Identity | |
| 34 | mobileNumber | 0-20 | O | N | Mobile number | |
| 35 | fax | 0-20 | O | AN | Fax | |
| 36 | countryName | 0-25 | O | AN | Country Name | |
| 37 | nationality | 0-50 | O | AN | Nationality | |
| 38 | suitNumber | 0-8 | O | AN | Suit Number | |
| 39 | accountType | 0-2 | O | AN | Account Type | |
| 40 | accountNo | 0-28 | O | AN | Account Number | |
| 41 | accountStatus | 0-2 | O | N | Account status | |
| 42 | embossName | 0-25 | O | ANS | Emboss name | |
| 43 | reqId | 1-8 | M | N | Id of each request | |
| 44 | isClearPANRequired | 1 | O | A | Y or N | If Y, clear PAN will present in response |
| 45 | isCVV2Reqired | 1 | O | A | Y or N | If Y, CVV2 will present in response |
| 46 | isSHA256PANRequired | 1 | O | A | Y or N | If Y, clear PAN hased with SHA256 in response |

**Response Format:**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | instCode | Institution Code | String | |
| 2 | cardLinkingRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |

| | | | | |
|---|---|---|---|---|
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of cardLinkingRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | cardLinkResp | List of response | Array | Array of Object |

**Array Elements of cardLinkResp Object for success response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | |
| 3 | respMsg | Response message | String | |
| 4 | actionCode | Action Code | String | |
| 5 | vpan | Vpan | String | |
| 6 | accountNo | Account Number | String | |
| 7 | pan | PAN | String | |
| 8 | clearPAN | Clear card number | String | Clear Pan value encrypted by Authman session key |
| 9 | cvv2 | Clear cvv2<br>Cvv2 value encrypted by Authman session key | String | Cvv2 value encrypted by Authman session key |

**Array Elements of cardLinkResp Object for failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | |
| 3 | respMsg | Response message | String | |
| 4 | actionCode | Validation or Execution | String | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/cardlink

Validation Request :

```
{
"instCode" : "FIS",
"reqMsgId" : "201951968545010001",
"cardLinkReq":[
{
 “reqId” : “201908271919300000”,
 “actionCode”:”V”,
  “vpan” :”MYMC000000000739”,
 “accountNo” :”0000000000045311 “,
 “pan” : “8040”,
“firstName” : “CUSTOMER”,
 “lastName” : “FIS”,
 “gender” : “F”,
 “marriedStatus” : “S”,
 “accountType” : “12”,
 “accountNo” : “0000000012020301”,
 “embossName” : “CORTEX”
}
]}
```

Validation Response:

```
{
    “respCode”: “901”,
    “respMsg”: “Request Processed.”,
    “processCode”: 115,
    “instCode”: “FIS”,
    “reqMsgId” : “201951968545010001”,
    “cardLinkingRes”: {
    “cardLinkResp”: [
     {
        “reqId”: “201908271919300000”,
        “respCode”: “00”,
        “respMsg”: “Validation done – Given Card is allowed to link with new card”,
        “actionCode”: “V”,
        “vpan”: “MYMC000000000739”,
        “accountNo”: “0000000000045311 “,
        “pan”: “8040”
     }
  ]
}
```

Execution Request :

```
{
“instCode” : “FIS”,
“reqMsgId” : “201951968545010001”,
“cardLinkReq”:[
{
 “reqId” : “201908271919300000”,
 “actionCode”:”E”,
  “vpan” :”MYMC000000000739”,
 “accountNo” :”0000000000045311 “,
 “pan” : “8040”,
“accountType” : “12”,
 “accountNo” : “0000000012020301”,
 “embossName” : “CORTEX”
}
]}
```

Execution Response :

```
{
    “respCode”: “901”,
    “respMsg”: “Request Processed.”,
    “processCode”: 115,
    “instCode”: “FIS”,
    “reqMsgId” : “201951968545010001”,
    “cardLinkingRes”: {
    “cardLinkResp”: [
      {
         “reqId”: “201908271919300000”,
         “respCode”: “00”,
         “respMsg”: “Card Linking Successful”,
         “actionCode”: “E”,
         “vpan”: “MYMC000000000739”,
         “accountNo”: “0000000000045311 “,
         “pan”: “8040”
      }
   ]
}
```

Failure Response :

```
{
    “respCode”: “901”,
    “respMsg”: “Request Processed.”,
    “processCode”: 115,
    “instCode”: “FIS”,
    “reqMsgId” : “201951968545010001”,
    “cardLinkingRes”: {
    “cardLinkResp”: [
     {
         “reqId”: “201908271919300000”,
         “respCode”: “01”,
         “respMsg”: “Given Card – Account is not Linked”,
         “actionCode”:”E”

      },
      {
         “reqId”: “201908271919300001”,
         “respCode”: “01”,
         “respMsg”: “Given Card – Account is not Linked”,
         “actionCode”:”E”

      }
   ]
}
```

## 11. Card Upgrade [Card Product Change]

Card product change is used to change a card's card product (or card type). The change will happen only the Cortex system and no change in the physical card.

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | cardProductChgReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of cardProductChgReq Object:**

| S. No | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | reqId | 18 | M | N | Id of each request | |
| 2 | actionCode | 1 | M | A | Validation or Execution | |
| 3 | vpan | 15 - 19 | M | AN | Unique identifier of card | |
| 4 | cardProduct | 4 - 4 | M | AN | Card Product | |

**Response Format:**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | instCode | Institution Code | String | |
| 2 | cardProductChngRes | JSON Object | JSON Object | |
| 3 | processCode | Process id of each request | Number | |
| 4 | reqMsgId | Request Message Id | String | |

| | | | | |
|---|---|---|---|---|
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of cardProductChngRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | cardProductChgResp | List of response | Array | Array of Object |

**Array Elements of cardProductChgResp Object for success response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | Stirng | |
| 2 | respCode | Response Code | Stirng | |
| 3 | respMsg | Response message | Stirng | |
| 4 | actionCode | Action Code | Stirng | |
| 5 | vpan | Vpan | Stirng | |
| 6 | cardProduct | Card Product | Stirng | |

**Array Elements of cardProductChgResp Object for failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | reqId | Request id of each request | Stirng | |
| 2 | respCode | Response Code | Stirng | |
| 3 | respMsg | Response message | Stirng | |
| 4 | actionCode | String | Validation or Execution | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/cardproductchng

Validation Request :

```
{
“instCode” : “MYNT”,
“reqMsgId” : “201951968545010001”,
“cardProductChgReq”:[
{
 “reqId” : “201908271919300000”,
 “actionCode”:”V”,
  “vpan” :”MYMC000000000143”,
 “cardProduct” :”MYMC”
},
{
 “reqId” : “201908271919300001”,
 “actionCode”:”V”,
  “vpan” :”MYMC000000000143”,
 “cardProduct” :”MYMC”
}
]}
```

Validation Response :

```
{
   “respCode”: “901”,
 “respMsg”: “Request Processed.”,
“processCode”: 114,
   “instCode”: “MYNT”,
“reqMsgId” : “201951968545010001”,
   “reqId”: “201908271919300045”,
   “cardProductChngRes”: {
 “cardProductChgResp”: [
       {
          “reqId”: “201908271919300000”,
          “respCode”: “00”,
          “respMsg”: “Validation done – Given card is allowed to update the card product”,
          “actionCode”: “V”,
          “vpan”: “MYMC000000000143”,
          “cardProduct”: “MYMC”
       },
       {
          “reqId”: “201908271919300001”,
          “respCode”: “00”,
          “respMsg”: “Validation done – Given card is allowed to update the card product”,
          “actionCode”: “V”,
          “vpan”: “MYMC000000000143”,
          “cardProduct”: “MYMC”
       }
    ]
}
```

Execution Request :

```
{
“instCode” : “MYNT”,
“reqMsgId” : “201951968545010001”,
“cardProductChgReq”:[
{
 “reqId” : “201908271919300000”,
 “actionCode”:”E”,
  “vpan” :”MYMC000000000143”,
 “cardProduct” :”MYMC”
},
{
 “reqId” : “201908271919300001”,
 “actionCode”:”E”,
  “vpan” :”MYMC000000000143”,
 “cardProduct” :”MYMC”
}
]}
```

Execution Response :

```
{
   “respCode”: “901”,
 “respMsg”: “Request Processed.”,
“processCode”: 114,
   “instCode”: “MYNT”,
“reqMsgId” : “201951968545010001”,
   “cardProductChngRes”: {
 “cardProductChgResp”: [
       {
```

```
      “reqId”: “201908271919300000”,
      “respCode”: “00”,
      “respMsg”: “Card Product Updated”,
      “actionCode”: “E”,
      “vpan”: “MYMC000000000143”,
      “cardProduct”: “MYMC”
    },
    {
      “reqId”: “201908271919300001”,
      “respCode”: “00”,
      “respMsg”: “Card Product Updated”,
      “actionCode”: “E”,
      “vpan”: “MYMC000000000143”,
      “cardProduct”: “MYMC”
    }
  ]
}
```

Failure Response :

```
{
  “respCode”: “901”,
 “respMsg”: “Request Processed.”,
“processCode”: 114,
  “instCode”: “MYNT”,
 “reqMsgId” : “201951968545010001”,
  “cardProductChngRes”: {
  “cardProductChgResp”: [
    {
      “reqId”: “201908271919300000”,
      “respCode”: “01”,
      “respMsg”: “Can’t find vpan:[MYMC000000000144]”,
      “actionCode”:”E”

    },
    {
      “reqId”: “201908271919300001”,
      “respCode”: “01”,
      “respMsg”: “Can’t find vpan:[MYMC000000000144]”,
      “actionCode”:”E”

    }
  ]
}
```

Failure Res:

```
{
  “respCode”: “901”,
 “respMsg”: “Request Processed.”,
“processCode”: 114,
  “instCode”: “MYNT”,
“reqMsgId” : “201951968545010001”,
  “cardProductChngRes”: {
  “cardProductChgResp”: [
    {
      “reqId”: “201908271919300000”,
      “respCode”: “01”,
      “respMsg”: “Crdproduct [MYMD] not found”,
       “actionCode”:”E”

    },
    {
      “reqId”: “201908271919300001”,
      “respCode”: “01”,
      “respMsg”: “Crdproduct [MYMD] not found”,
       “actionCode”:”E”

    }
  ]
}
```

## 12. Customer details enquiry

Customer Enquiry service used for getting information about customer. Clearn PAN or CVV2 will get in response only when customer details fetch using VPAN/PAN.

Request Format :

| S. No | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | pFlag | 1 | M | A | S | If ‘S’, system will pocess synchronously else Deafult set ‘S’ [It always synchronous service] |
| 3 | reqId | 18 | M | N | Id of each request | |
| 4 | custCode | 0 - 10 | C | AN | Customer Code – it’s unique identifier for each customer. | |
| 5 | accNo | 1 - 28 | C | AN | Account number. | |
| 6 | accCurr | 3 | C | AN | Account currency number. It’s three digits. | |
| 7 | clearPan | 19 | C | AN | Clear Card Number. | |
| 8 | vpan | 15 - 19 | C | AN | Virtual Card Number / PAN ALIAS | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | seqNo | 1 | O | N | Sequence Number | |
| 10 | idNumber | 0 - 12 | C | AN | Id Number (like passport, Driving License, ..etc) | |
| 11 | mobNo | 0 - 20 | C | AN | Mobile Number | |
| 12 | eMail | 0 - 64 | C | AN | e-Mail ideentifiation | |
| 13 | firstName | 0 - 50 | C | AN | Customer First Name | |
| 14 | lastName | 0 - 50 | C | AN | Customer Last Name | |
| 15 | reqMsgId | 18 | M | String | Request Message Id | |
| 16 | startRecordNo | 2 | M | N | Start Record Number | |
| 17 | endRecordNo | 2 | M | N | End Record Number | |
| 18 | isClearPANRequired | 1 | O | A | Y or N | If Y, clear PAN will present in response |
| 19 | isCVV2Reqired | 1 | O | A | Y or N | If Y, CVV2 will present in response |
| 20 | isSHA256PANRequired | 1 | O | A | Y or N | If Y, clear PAN hased with SHA256 in response |

**Note:** Any of the Customer code, Account Number, Clear PAN, Virtual PAN, ID number, Mobile number, email id or "Firstname + Lastname of customer" is mandatory in the request. If none of the field is given, the request will be rejected.

**Response Format:**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | instCode | Institution Code | AN | |
| 2 | customerEnquiryRes | Response Object | Object | Json Object |
| 3 | reqId | Request id of each request | AN | |
| 4 | processCode | Number | Process id of each request | |
| 5 | reqMsgId | String | Request Message Id | |

**Elements of customerEnquiryRes Object for success response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | custDetailsResp | List of response | Array | Array of Object |
| 2 | reqId | Request Id | String | |
| 3 | respCode | Response Code | String | |
| 4 | respMsg | Response Message | String | |
| | | | | |

**Array Elements of custDetailsResp Object Response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | accountDetailsResp | List of account Response | Array | It have list of account with cards |
| 2 | custCode | Customer Code | AN | |
| 3 | extCustCode | Ext Customer Code | AN | |
| 4 | typeId | Type Id of Customer | N | |
| 5 | title | Titles of customers | AN | Mr, Mrs, etc.. |
| 6 | lastName | Last name of the customer | AN | |
| 7 | firstName | First name of the customer | AN | |
| 8 | dateOfBirth | Customer date of birth | AN | Format – YYYYMMDD |
| 9 | married | Marital Status | AN | |
| 10 | gender | Gender | AN | |
| 11 | profileCode | Profile Code | N | |
| 12 | mobNo | Moblie Number | AN | |
| 13 | emailId | Customer Email | AN | |
| 14 | addressLine0 | Address Line 0 | AN | |
| 15 | addressLine1 | Address Line 1 | AN | |
| 16 | addressLine2 | Address Line 2 | AN | |

| | | | | |
|---|---|---|---|---|
| 17 | addressLine3 | Address Line 3 | AN | |
| 18 | homeCity | Home City | AN | |
| 19 | homeCountry | Home Country | AN | Home State/Province |
| 20 | homePostCode | Home Postal Code | AN | |
| 21 | homeCountryCode | Home Country Code | N | |
| 22 | homeTel | Home Telephone Number | AN | |
| 23 | workAddRLine1 | Work Address Line 1 | AN | |
| 24 | workAddRLine2 | Work Address Line 2 | AN | |
| 25 | workAddRLine3 | Work Address Line 3 | AN | |
| 26 | workCity | Work City | AN | |
| 27 | workCountry | Work Country | AN | Work State/Province |
| 28 | workPostCode | Work Postal Code | AN | |
| 29 | workCountyCode | Work Counrty Code | N | |
| 30 | workTel | Work Telephone Number | AN | |
| 31 | userData1 | | AN | |
| 32 | userData2 | | AN | |
| 33 | userData3 | | AN | |
| 34 | pobox | | AN | |
| 35 | nationalId | National Identiy Number | AN | |
| 36 | idNumber | Id Number | AN | |

**Array Elements of accountDetailsResp object response:**

| S. No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | accNo | Account Number | AN | |
| 2 | currencyCode | Currency Code | AN | |
| 3 | accountType | Account Type | AN | |
| 4 | accountStatus | Account Status | AN | |
| 5 | avlBalance | Available Balance with decimal | AN | |
| 6 | blkAmount | Block Amount with decimal | AN | |
| 7 | otb | OpenToBuy (AVLBAL-BLKAMT) | AN | |

| | | | | |
|---|---|---|---|---|
| 8 | branchCode | Branch Code | AN | |
| 9 | cardDetailsResp | List of cards details | Array | |

**Array Elements of cardDetailsResp Object response;**

| S. No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | cardProduct | Card Product | AN | |
| 2 | oldCardProduct | Old Card Product | AN | |
| 3 | clearPan | Clear Card Number value encrypted by Authman session key. | AN | It's optional<br>Clear Card Number value encrypted by Authman session key |
| 4 | vpan | Vitural Pan | AN | |
| 5 | panDisplay | Pan Display | AN | |
| 6 | expDate | Expiry Date | N | It's optional<br>Expiry Date value encrypted by Authman session key |
| 7 | oldExpDate | Old Expiry Date | N | |
| 8 | cardStatus | Card Status | N | |
| 9 | cardsTitle | Card Holder Title | AN | |
| 10 | cardLastName | Last Name of Card | AN | |
| 11 | cardFirstName | Fisrt Name of Card | AN | |
| 12 | embossName | Card Emboss Name | AN | |
| 13 | corpFlag | Corporate Flag Indicator | AN | 0 or empty – Normal / General Customrt<br>1 – Corporate Customer<br>2 – Corporate Master |
| 14 | cvv2 | Cvv2 value encrypted by Authman session key | N | It's optional.<br>Cvv2 value encrypted by Authman session key |
| 15 | cardStatusDesc | Card Status Desc | AN | Status Desctiption |
| 16 | cardProductDesc | Card Product Description | AN | |
| 17 | oldCardProductDesc | Old Card Product Description | AN | |

**Array Elements of customerEnquiryRes Object for failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | **99** |
| 3 | respMsg | Response message | String | **No Data** |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/getcustomerdetails

Sample Request:

{

```
“instCode”:”CLNT”,
“reqMsgId” : “201951968545010001”,
“pFlag”:”s”,
“reqId”:”201908271919300000”,
“vpan” : “PMVC000000000299”,
“startRecordNo” : “1”
“endRecordNo” : “10”
}
```

Success Response:

```
{
   “processCode”: 208,
   “instCode”: “CLNT”,
   “reqMsgId”: “201951968545010001”,
   “reqId”: “201908271919304171”,
   “customerEnquiryRes”: {
      “reqId”: “201908271919304171”,
      “respCode”: “00”,
      “respMsg”: “SUCCESS”,
     “totalRecord” : “20”,
      “custDetailsResp”: [
         {
            “custCode”: “000000000007”,
            “extCustCode”: “000000100829”,
            “typeId”: “1”,
            “title”: “ “,
            “lastName”: “CLNT”,
            “firstName”: “CUSTOMER”,
            “dateOfBirth”: “20191130”,
            “married”: “M”,
            “gender”: “F”,
            “profileCode”: “0”,
            “mobNo”: “0000009187654312”,
            “emailId”: “ “,
            “addressLine0”: “ “,
            “addressLine1”: “ “,
            “addressLine2”: “ “,
            “addressLine3”: “ “,
            “homeCity”: “ “,
            “homeCountry”: “ “,
            “homePostCode”: “ “,
            “homeCountryCode”: “   “,
            “homeTel”: “ “,
            “workAddRLine1”: “ “,
            “workAddRLine2”: “ “,
            “workAddRLine3”: “ “,
            “workCity”: “ “,
            “workCountry”: “ “,
            “workPostCode”: “ “,
            “workCountyCode”: “   “,
            “workTel”: “ “,
  “userData1”: “1111111111”,
  “userData2”: “1111111111”,
  “userData3”: “ “,
  “accountDetailsResp”: [
    {
      “accNo”: “0009008761109112”,
      “currencyCode”: “608”,
      “accountType”: “12”,
      “accountStatus”: “00”,
      “avlBalance”: “0”,
      “blkAmount”: “0”,
      “otb”: “0”,
      “branchCode”: “0001”,
      “cardDetailsResp”: [
        {
          “cardProduct”: “MAMX”,
          “oldCardProduct”: “”,
          “clearPan”: “ “,
          “vpan”: “CLNT00000001094”,
          “panDisplay”: “375190_5531”,
          “expDate”: “20241130”,
          “oldExpDate”: “22630831”,
          “cardStatus”: “00”,
          “cardsTitle”: “ “,
          “cardLastName”: “CUSTOMER”,
          “cardFirstName”: “CLNT”,
          “embossName”: “ “,
          “corpFlag”: “0”,
          “cvv2”: “ “
        }
      ]
    },
    {
      “accNo”: “0000009187654312”,
      “currencyCode”: “608”,
      “accountType”: “12”,
      “accountStatus”: “00”,
      “avlBalance”: “0”,
      “blkAmount”: “0”,
      “otb”: “0”,
      “branchCode”: “0001”,
      “cardDetailsResp”: [
        {
          “cardProduct”: “MAMX”,
          “oldCardProduct”: “”,
          “clearPan”: “ “,
          “vpan”: “CLNT00000001086”,
          “panDisplay”: “375190_5523”,
          “expDate”: “20241130”,
          “oldExpDate”: “22630831”,
                        “cardStatus”: “00”,
                        “cardsTitle”: “ “,
                        “cardLastName”: “CUSTOMER”,
                        “cardFirstName”: “CLNT”,
                        “embossName”: “ “,
                        “corpFlag”: “0”,
                        “cvv2”: “ “
                     }
                  ]
               }
            ]
         }
      ]
   }
}
```

Failure Response:

```
{
   “processCode”: 208,
   “instCode”: “CLNT”,
   “reqMsgId”: “201951968545010001”,
   “reqId”: “201908271919300000”,
   “customerEnquiryRes”: {
      “reqId”: “201908271919300000”,
      “respCode”: “01”,
      “respMsg”: “NODATA”
   }
}
```

## 13. Virtual Card Creation

Virtual card creation API used to create virtual card in cortex.

**Request Format:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | virtualCardReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of virtualCardReq Object:**

| S. No | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 1 | M | A | Validation or Execution | V or E |
| 2 | branchCode | 1-8 | M | AN | Branch Code | |
| 3 | cardProduct | 1-4 | M | AN | Card Product | MAMX(for amex card) |
| 4 | custCode | 0-10 | M | AN | Customer Code | |
| 5 | custType | 0-5 | O | N | Customer Type | |
| 6 | title | 0-4 | O | ANS | Title of customer | |
| 7 | firstName | 2-50 | M | ANS | Customer’s first name | |
| 8 | lastName | 1-50 | M | ANS | Customer’s last name | |
| 9 | birthDate | 0-8 | O | Date | Birth date | |
| 10 | gender | 1 - 1 | M | A | Gender | |
| 11 | marriedStatus | 1 - 1 | M | A | Marital Status | |
| 12 | emailId | 0-64 | O | AN | Email Id | |
| 13 | profession | 0-5 | O | N | Profession | |
| 14 | homeAddr1 | 0-35 | C | ANS | Home Address 1 | Work address or Home address is mandatory |
| 15 | homeAddr2 | 0-35 | C | ANS | Home Address 2 | Work address or Home address is mandatory |
| 16 | homeAddr3 | 0-35 | C | ANS | Home Address 3 | Work address or Home address is mandatory |
| 17 | homeCity | 0-40 | C | ANS | Home City | |
| 18 | homeCounty | 0-20 | C | AN | Home State/Province | |
| 19 | homeCountry | 0-3 | C | N | Home Country Code | |
| 20 | homeTel | 0-20 | C | N | Home Telephone | |
| 21 | homePostalCode | 0-10 | C | AN | Home postal code | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 22 | poBox | 0-8 | O | AN | Po Box | |
| 23 | workAddr1 | 0-30 | C | AN | Work address 1 | Work address or Home address is mandatory |
| 24 | workAddr2 | 0-30 | C | AN | Work address 2 | Work address or Home address is mandatory |
| 25 | workAddr3 | 0-30 | C | AN | Work address 3 | Work address or Home address is mandatory |
| 26 | workCity | 0-40 | C | AN | Work city | |
| 27 | workCounty | 0-20 | C | AN | Work State/Province | |
| 28 | workCountry | 0-3 | C | N | Work Country Code | |
| 29 | workTel | 0-20 | C | N | Work telephone | |
| 30 | workPostalCode | 0-10 | C | AN | Work postal code | |
| 31 | identificationNumber | 0-12 | O | AN | Id number | |
| 32 | nationalId | 0-22 | O | AN | National Identity | |
| 33 | mobileNumber | 0-20 | O | N | Mobile number | |
| 34 | fax | 0-20 | O | AN | Fax | |
| 35 | countryName | 0-25 | O | AN | Country Name | |
| 36 | nationality | 0-50 | O | AN | Nationality | |
| 37 | suitNumber | 0-8 | O | AN | Suit Number | |
| 38 | accountType | 0 - 2 | O | AN | Account Type | |
| 39 | accountNo | 0 - 28 | O | AN | Account Number | |
| 40 | accountStatus | 0-2 | O | N | Account status | |
| 41 | reqId | 1-8 | M | N | Id of each request | |
| 42 | pan | 17-19 | O | AN | Pan or clear pan | Card number |
| 43 | isClearPANRequired | 1 | O | A | Y or N | If Y, clear PAN will present in response |
| 44 | isCVV2Reqired | 1 | O | A | Y or N | If Y, CVV2 will present in response |
| 45 | isSHA256PANRequired | 1 | O | A | Y or N | If Y, clear PAN hased with SHA256 in response |

Note : Work address details or Home address details any one should mandatory

**Response Format:**

| S.No. | Field | Description | Comments |
|---|---|---|---|
| 1 | instCode | Institution Code | |
| 2 | virtualCardRes | JSON Object | JSON Object |
| 3 | processCode | Number | Process id of each request |
| 4 | reqMsgId | String | Request Message Id |
| 5 | respCode | String | Response Code |
| 6 | respMsg | String | Response message |

**Elements of virtualCardRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | virtualCardResp | List of response | Array | Array of Object |

**Array Elements of virtualCardResp Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| **1** | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | clrPan | String | Clear pan encrypted by Authman session key. | It will be generated only for execution request not for validation.<br>Clear pan encrypted by Authman session key. |
| 5 | vpan | String | Vpan | It will be generated only for execution request not for validation. |
| 6 | panDispaly | String | Pan Display | It will be generated only for execution request not for validation. |
| 7 | encryptedPan | String | Ecncryted Pan | It will be generated only for |

| | | | | |
|---|---|---|---|---|
| | | | | execution request not for validation. |
| 8 | cvv2 | String | Cvv2 value encrypted by Authman session key | Cvv2 value encrypted by Authman session key |
| 9 | expDate | String | Expiry Date<br>Encrypted by Authman session key. | It will be generated only for execution request not for validation.<br>Expiry Date encrypted by Authman session key. |
| 10 | actionCode | String | Action code | |
| 11 | branchCode | String | Branch Code | |
| 12 | cardProduct | String | Card Product | |
| 13 | firstName | String | First Name | |
| 14 | lastName | String | Last Name | |
| 15 | gender | String | Gender | |
| 16 | marriedStatus | String | Marital Status | |
| 17 | accountType | String | Account Type | |
| 18 | accountNo | String | Account Number | |

**Array Elements of virtualCardResp Object for failure response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | Only when error response received |
| 4 | actionCode | String | Validation or Execution | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/virtualcardcreation

Validation Request :

```
{
  "instCode": "FNMG",
  "reqMsgId": "202051968545010004",
  "pFlag": "E",
  "virtualCardReq": [{
    "reqId": "202008271919300003",
    "custCode":"custt0003447",
    "homeAddr1":"As3",
    "homeAddr2":"Akilandeshwari aptmt",
    "homeAddr3":"cart track road velachery",
    "homeCity":"chennai",
    "homeCounty":"Tamilnadu",
    "homeCountry":"IND",
    "homeTel":"0444232456",
    "homePostalCode":"636105",
    "actionCode": "V",
    "branchCode": "0027",
    "cardProduct": "0123",
    "firstName": "Fincarecustomerone",
    "lastName": "Fincareone",
    "gender": "F",
    "marriedStatus": "S",
    "accountType": "10",
    "accountNo": "0000000012020520",
    "pan":"6087231299231131"
  }]
}
```

Execution Response :

```
{
  "respCode": "901",
  "respMsg": "Request Processed.",
  "processCode": 215,
  "instCode": "FNMG",
  "reqMsgId": "202051968545010004",
  "reqId": "202008271919300003",
  "virtualCardRes": {
    "virtualCardResp": [
      {
        "reqId": "202008271919300003",
        "respCode": "00",
        "vpan": "FNMG000000001005",
        "panDispaly": "608723_1131",
        "expDate": "2025-03-31",
        "cvv2": "234",
        "actionCode": "000",
        "cardProduct": "0123",
        "firstName": "Fincarecustomerone",
        "lastName": "Fincareone",
        "accountType": "10",
        "accountNo": "0000000012020520"
      }
    ]
  }
}
```

Failure Response :

```
{
"respCode": "904",
"respMsg": "Missing the CustCode in Customer Details  : CustCode",
"processCode": 215,
"instCode": "FNMG",
"reqMsgId": "202051968545010005",
"reqId": "202008271919300004"
}
```

## 14. Virtual to physical card creation

Virtual to physical card creation API used to convert virtual card to physical card creation. User will get physical card for him / her virtual card.

**Request Format:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | virtual2PhysicalReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of virtual2PhysicalReq Object:**

| S. No | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 1 | M | A | Validation or Execution | V or E |
| 2 | custCode | 0-10 | O | AN | Customer Code | |
| 3 | photoId | 1-4 | O | AN | Photo ID | |
| 4 | aadharNo | 0 -2 | O | N | Aadhar card Number | |
| 5 | pan | 17-19 | M | AN | Pan or clear pan | |
| 6 | reqId | 18 | M | N | Id of each request | |
| 7 | persoBureaoCode | 1-8 | M | AN | Perso Bureao Code | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | cardProduct | 1-4 | M | AN | Card Product | |
| 9 | orderId | 1-20 | O | AN | Order Id | |
| 10 | awbNo | 1-20 | O | AN | Air Waybill Number | |
| 11 | qrCode | 1-20 | O | AN | QR Code | |
| 12 | courierVendor | 1-20 | O | AN | Courier Vendor | |

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | virtual2PhysicalRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of virtual2PhysicalRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | virtual2PhysicalResp | List of response | Array | Array of Object |

**Array Elements of virtual2PhysicalResp Response Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | actionCode | String | Validation or Execution | |
| 5 | pan | String | Clear or Encrypted Pan<br>Pan / Clear Pan value encrypted by Authman session key. | Pan / Clear Pan value encrypted by Authman session key. |
| 6 | custCode | String | Customer Code | |

**Array Elements of virtual2PhysicalResp Response Object for failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | |
| 3 | respMsg | Response message | String | |
| 4 | actionCode | Validation or Execution | String | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/virtual2physical

Execution Request:

```
{
"instCode": "FNMG",
"reqMsgId": "202051968545010011",
"virtual2PhysicalReq": [{
"reqId": "202008271919300011",
"actionCode": "V",
"pan": "6087231299231142",
"persoBureaoCode": "FCT",
"cardProduct": "PDC1"
}]
}
```

Exceution Response:

```
{
  "respCode": "901",
  "respMsg": "Request Processed.",
  "processCode": 216,
  "instCode": "FNMG",
  "reqMsgId": "202051968545010011",
  "reqId": "202008271919300011",
  "virtual2PhysicalRes": {
    "virtual2PhysicalResp": [
      {
        "reqId": "202008271919300011",
        "respCode": "00",
        "actionCode": "000",
        "pan": "6087231299231142",
        "custCode": "custt0003449"
      }
    ]
  }
}
```

## 15. Adjust Account Balance

Adjust account balance API used to account debit and account credit for all purchase or fun transfer.

**Request Format:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | adjAccountBlanceReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of adjAccountBlanceReq Object:**

| S. No | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 1 | M | A | Validation or Execution | V or E |
| 2 | reqId | 18 | M | N | Id of each request | |
| 3 | pan | 17-19 | C | AN | Pan number | |
| 4 | vpan | 15 - 19 | C | AN | VPAN | |
| 5 | accNo | 1 - 28 | C | AN | Account number. | |
| 6 | accCurr | 3 | C | AN | Account currency number. It's three digits. | |
| 7 | hostName | 1-4 | M | AN | Host Name | |
| 8 | fiid | 1-11 | M | AN | | Forwarding Institution ID |
| 9 | inDateRec | 10-10 | M | N | | |
| 10 | trn | 1-64 | M | AN | | |
| 11 | hostCmd | 1-10 | M | N | | Command to adjust balance, default – 1000 |
| 12 | amtBill | 1-6 | M | N | Amount of bill | |
| 13 | Narrative | 1-128 | M | AN | | |
| 14 | User | 1-15 | M | AN | | |
| 15 | hostTimeStamp | 1-17 | M | AN | Datetimeformat with miliseconds | yyyyMMddHHmmssSSS<br>Sample:<br>20201204203332000 |

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | adjAccountBlanceRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of adjAccountBlanceRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | adjAccountBlanceResp | List of response | Array | Array of Object |

**Array Elements of adjAccountBlanceResp Response Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | actionCode | String | Validation or Execution | |
| 5 | accNo | String | Account Number | |
| 6 | accCurr | String | Account Currency | |
| 7 | FIID | Stirng | | |
| 8 | TRN | String | | |
| 9 | avlBal | String | Available Amount | |
| 10 | blkAmt | String | Block Amount | |

**Array Elements of adjAccountBlanceResp Response Object for failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | |
| 3 | respMsg | Response message | String | |
| 4 | actionCode | Validation or Execution | String | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/adjaccountbalance

Validation Request :

```
{
        "instCode" : "IEBT",
        "pFlag":"s",
        "reqMsgId": "202011191341301254",
        "adjAccountBlanceReq":[
        {
        "reqId" : "202011191341301254",
        "actionCode":"E",
        "pan":"6161610008671396",
        "accNo":"100000874001",
        "accCurr":"356",
        "hostName":"IST",
        "hostCmd":"1000",
        "inDateRec":"20201214",
        "fiid":"212123237",
        "narrative":"Deducting FEE",
        "trn":"DD08201918085457",
        "hostTimeStamp":"20201214203322000",
        "amtBill":"10",
        "user":"WEB"
        }]
}
```

Validation Response:

```
{
  "respCode": "901",
  "respMsg": "Request Processed.",
  "processCode": 220,
  "instCode": "IEBT",
  "reqMsgId": "202011191341301254",
  "reqId": "202011191341301254",
  "adjAccountBlanceRes": {
    "adjAccountBlanceResp": [
      {
        "reqId": "202011191341301254",
        "respCode": "00",
        "respMsg": "Balance Adjustment successful",
        "actionCode": "3",
        "accNo": "100000874001",
        "accCurr": "356",
        "avlBal": "56041.00000000000",
        "blkAmt": "0.000000",
        "fiid": "212123237"
      }
    ]
  }
}
```

Execution Request :

```
        {
                "instCode" : "IEBT",
                "pFlag":"s",
                "reqMsgId": "202011191341301254",
                "adjAccountBlanceReq":[
                {
                "reqId" : "202011191341301254",
                "actionCode":"E",
                "pan":"6161610008671396",
                "accNo":"100000874001",
                "accCurr":"356",
                "hostName":"IST",
                "hostCmd":"1000",
                "inDateRec":"20201214",
                "fiid":"212123237",
                "narrative":"Deducting FEE",
                "trn":"DD08201918085457",
                "hostTimeStamp":"20201214203322000",
                "amtBill":"10",
                "user":"WEB"
                }]
        }
```

Execution Response:

```
{
"respCode": "901",
"respMsg": "Request Processed.",
"processCode": 220,
"instCode": "IEBT",
"reqMsgId": "202011191341301254",
"reqId": "202011191341301254",
"adjAccountBlanceRes": {
"adjAccountBlanceResp": [
{
"reqId": "202011191341301254",
"respCode": "00",
"respMsg": "Balance Adjustment successful",
"actionCode": "3",
"accNo": "100000874001",
"accCurr": "356",
"avlBal": "56041.00000000000",
"blkAmt": "0.000000",
"fiid": "212123237"
}
]
}
}
```

## 16. Account Load

Account load API used to load amount in to user account. Based on account number and card details.

**Request Format:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| ss1 | instCode | 4 | M | AN | Institute Code | |
| 2 | accountLoadReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of accountLoadReq Object:**

| S. No | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | actionCode | 1 | M | A | Validation or Execution | V or E |
| 2 | reqId | 18 | M | N | Id of each request | |
| 3 | accNo | 1 - 28 | C | AN | Account number. | |
| 4 | accCurr | 3 | C | AN | Account currency number. It’s three digits. | |
| 5 | loadType | 1-99 | M | AN | Load Type | |
| 6 | loadSrc | 1-99 | M | AN | Load Source | CRDLOADSRC_UNKNOWN<br>0 /* unknown */<br>CRDLOADSRC_CRI<br>1 /* card request import */<br>CRDLOADSRC_WEBSRVC<br>2 /* general web service */<br>CRDLOADSRC_IVR<br>3 /* IVR */<br>CRDLOADSRC_WEBSITE<br>4 /* website */<br>CRDLOADSRC_PAYPOINT<br>5 /* Paypoint */<br>CRDLOADSRC_PAYZONE<br>6 /* Payzone */<br>CRDLOADSRC_POSTOFFICE<br>7 /* post office */<br>CRDLOADSRC_AGENT<br>8 /* agent */<br>CRDLOADSRC_HEADOFFICE<br>9 /* head office */<br>CRDLOADSRC_CALLCNTR<br>10 /* call centre */ |
| 7 | amtBill | 1-6 | M | N | Aount of bill | |
| 8 | extCode | 1-16 | M | N | Extention code | External Code – Unique reference number for each transaction.<br>16 digits maximum |
| 9 | agentId | 1-12 | M | AN | Agent Id | Card Acceptor ID |
| 10 | pan | 17-19 | C | AN | Pan number | |
| 11 | vpan | 15 - 19 | C | AN | VPAN | |
| 12 | action | 1-3 | M | N | Action codes | Action code 1, 2, 3, 4, 101<br>1 - Load Request<br>2 - Advice<br>3 - Cancel<br>4 - Verify |
| 13 | expDate | 4 | CO | AN | Card’s expiry date | YYMM |

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | accountLoadRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of accountLoadRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | accountLoadResp | List of response | Array | Array of Object |

**Array Elements of accountLoadResp Response Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | actionCode | String | Validation or Execution | |
| 5 | action | String | Action | |
| 6 | accNo | String | Account Number | |
| 7 | extCode | String | | |
| 8 | accCurr | String | Account Currency | |
| 9 | amtBill | String | Amount Bill | |
| 10 | avlBal | String | Available amount | |
| 11 | blkAmt | String | Block Amount | |
| 12 | preAuthBlk | String | | |
| 13 | preAuthUnclr | String | Pre auth unclear amount | |
| 14 | firstName | String | First Name | |
| 15 | lastName | String | Last Name | |
| 16 | title | String | Title | |

**Array Elements of accountLoadResp Response Object for failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | |
| 3 | respMsg | Response message | String | |
| 4 | actionCode | Validation or Execution | String | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/accountload

Validation Request :

```
{
      “instCode” : “FIS”,
     “reqMsgId” : “201951968545010001”,
      “accountLoadReq”:[{
              “reqId” : “201908271919300000”,
              “actionCode”:”V”,
              "accNo":"",
"accCurr":"",
"loadType":"",
"loadSrc":"",
"amtBill":"",
"extCode":"",
"agentId":"",
"pan":"",
"vpan":"",
"action":""
        }]
}
```

Validation Response:

```
{
       “respCode”: “901”,
       “respMsg”: “Request Processed.”,
        “processCode”: 218,
    “instCode”: “FIS”,
     “reqMsgId” : “201951968545010001”,
    “accountLoadRes”: {
     “accountLoadResp”: [
        {
            “reqId”: “201908271919300000”,
            “respCode”: “00”,
            “respMsg”: “Given card is allowed to card upgrade.”,
            “actionCode”: “V”,
"action":"",
"accNo":"",
"extCode":"",
"accCurr":"",
"amtBill":"",
"avlBal":"",
"blkAmt":"",
"preAuthBlk":"",
"preAuthUnclr":"",
"firstName":"",
"lastName":"",
"title":""
        },
        {
            “reqId”: “201908271919300001”,
            “respCode”: “01”,
            “respMsg”: “Already Processed –“
        }
    ]
}
```

Execution Request :

```
{
      “instCode” : “FIS”,
     “reqMsgId” : “201951968545010001”,
      “accountLoadReq”:[{
          “reqId” : “201908271919300000”,
          “actionCode”:”E”,
          "action":"",
"accNo":"",
"extCode":"",
"accCurr":"",
"amtBill":"",
"avlBal":"",
"blkAmt":"",
"preAuthBlk":"",
"preAuthUnclr":"",
"firstName":"",
"lastName":"",
"title":""
        }]
}
```

Execution Response:

```
{
       “respCode”: “901”,
       “respMsg”: “Request Processed.”,
        “processCode”: 218,
    “instCode”: “FIS”,
     “reqMsgId” : “201951968545010001”,
    “accountLoadRes”: {
     “accountLoadResp”: [
        {
            “reqId”: “201908271919300000”,
            “respCode”: “00”,
            “respMsg”: “Request Processed”,
            “actionCode”: “E”,
"action":"",
"accNo":"",
"extCode":"",
"accCurr":"",
"amtBill":"",
"avlBal":"",
"blkAmt":"",
"preAuthBlk":"",
"preAuthUnclr":"",
"firstName":"",
"lastName":"",
"title":""
        }
```

```
   ]
}
```

Failure Response:

```
{
      “respCode”: “901”,
      “respMsg”: “Request Processed.”,
       “processCode”: 218,
   “instCode”: “FIS”,
    “reqMsgId” : “201951968545010001”,
   “accountLoadRes”: {
    “accountLoadResp”: [
       {
          “reqId”: “201908271919300000”,
          “respCode”: “01”,
          “respMsg”: “Request not Processed - error”,
  }]}}
```

## 17. Set PIN

SetPIN API is to be used to set new PIN for a card. PIN should be sent as ZPK encrypted PIN block.

**Request Format:**

| S. No | Field | Min-Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institution Code | Example FIS |
| 2 | reqId | 18 | M | N | Id of each request, uniquely identified | |
| 3 | pan | 16-19 | C | Character | Clear card number (or number used for PIN block formation) | |
| 4 | vpan | 0-32 | C | Character | Virtual PAN (or) PAN Alias | |
| 5 | seqNo | 1 | M | Number | PAN sequence no | Card Sequence Number – Default – 0 "seqNo": "0", |
| 6 | keyRef | 16 | M | Character | ZPK Reference Code | Example FISZPK<br>Key Reference – Default - LOCAL |

| | | | | | | |
|---|---|---|---|---|---|---|
| 7 | encryptedPin | 16 | M | String | Encrypted PIN | |
| 8 | pinBlockFormat | 1 | M | Number | Pin Block format | 01 – ANSI format<br>03 – PIN pad or Diebold format |

**Response Format**:

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | instCode | Institution Code | String | |
| 2 | reqId | Request id of each request | String | |
| 3 | respCode | Response Code | String | |
| 4 | respMsg | Response message | String | |
| 5 | vpan | Vpan | String | Based on req it will pe pan or vpan |
| 6 | seqNo | PAN sequence no | String | |
| 7 | keyRef | PAN | String | |
| 8 | encryptedPin | ZPK Reference Code | String | |
| 9 | pinBlockFormat | Pin Block format | String | |

**Failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | |
| 3 | respMsg | Response message | String | |
| 4 | instCode | Institution Code | String | |

**Note: Either PAN or VPAN should be given as input.**

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/setpin

Set Pin Request :

```
{          “instCode” : “FIS”,
        “reqId”: “201908271919300000”,
        “ vpan “: “ PMMG000000000012 “,
        “ seqNo “: “0”,
        “ keyRef “: “ PORTAL”,
        “ encryptedPin “: “ E807FBE63941CEF6”,
        “ pinBlockFormat “: “1”
}
```

Set Pin Response:

```
{

        “instCode”: “FIS”,
        “reqId”: “201908271919300000”,
        “respCode”: “00”,
        “respMsg”: “Set PIN request successful”,
        “ vpan “: “ PMMG000000000012 “,
        “ seqNo “: “0”,
        “ keyRef “: “ PORTAL”,
        “ encryptedPin “: “ E807FBE63941CEF6”,
        “ pinBlockFormat “: “1”

}
```

Failure Response :

```
{
  “instCode”: “FIS”,
  “reqId”: “201908271719300000”,
  “respCode”: “01”,
  “respMsg”: “PAN – VPAN mismatch in request”
}
```

## 18. Change PIN

Change PIN is used to change PIN for a card. Both current PIN and new PIN should be sent as ZPK encrypted PIN block.

**Request Format**:

| S. No | Field | Min-Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institution Code | |
| 2 | reqId | 18 | M | N | Id of each request | |
| 3 | oldPinBlock | 16 | M | AN | Old PIN (current PIN) block | |
| 4 | newPinBlock | 16 | M | AN | New pin block | |
| 5 | expDate | 4 | M | AN | Card’s expiry date | YYMM |
| 6 | localDate | 0-8 | O | AN | Current Date | YYYYMMDD |
| 7 | keyReferenceCode | 1-16 | M | AN | Key Reference Code | Example FISZPK |
| 8 | pinBlockFormat | 1-2 | M | AN | Pin Block Format | 01 – ANSI format<br>03 – PIN pad or Diebold format |
| 9 | pan | 16-19 | M | AN | Clear Card Number | |
| 10 | afe | 4 | M | AN | Financial Entity | |
| 11 | termSeq | 4 | M | N | Term Seq number | It should be unique for each request.<br>Transaction Sequence Number from Terminal |

**Response Format**:

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | instCode | Institution Code | String | |
| 2 | reqId | Request id of each request | String | |
| 3 | respCode | Response Code | String | **Succes - 000** |

**Failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | |
| 3 | respMsg | Response message | String | |
| 4 | instCode | Institution Code | String | |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/changepin

Change Pin Request :

```
{
“instCode” : “FIS”,
“afe”:”FIS”,
“termSeq”:”5315”,
“oldPinBlock”:”9A4FBC381298D0CE”,
“newPinBlock”:”FD5E3EFDC38FAA33”,
“expDate”:”2404”,
“localDate”:”20191019”,
“keyReferenceCode”:”LOCAL”,
“pinBlockFormat”:”1”,
“pan”:”D26E0C17293AC8D56EE”,
“reqId”: “201912011919310023”
}
```

Change Pin Response:

```
{
   “instCode”: “FIS”,
   “reqId”: “201912161534069147”,
   “respCode”: “000”
}
```

Failure Response :

```
{
   “instCode”: “FIS”,
   “reqId”: “201912011919310023”,
   “respCode”: “13”,
   “respMsg”: “Could not get Transaction from TLOG !”
}
```

## 19. Verify PIN

PIN can be verified using this API for a card.

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | verifyPinReq | 1-50 | M | Array | List of requests | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of verifyPinReq Object:**

| S. No | Field | Max Length | Mandatory ? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | reqId | 18 | M | N | Id of each request | |
| 2 | actionCode | 1 | M | A | Validation or Execution | |
| 3 | vpan | 15 - 19 | CO | AN | Unique identifier of card | |
| 4 | pan | 17-19 | CO | N | Pan Number | |
| 5 | encPin | 30-40 | M | AN | Encrypted Pin | |
| 6 | crdPinTyp | 0-1 | O | N | Card Pin Type | Allow 0 to 9 only<br>0 – is default |
| 7 | whoSet | 0-30 | O | AN | | |

**Response Format:**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | instCode | Institution Code | String | |
| 2 | verifyPinRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process id of each request | |
| 4 | reqMsgId | String | Request Message Id | |

| | | | | |
|---|---|---|---|---|
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of verifyPinRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | **verifyPinResp** | List of response | Array | Array of Object |

**Array Elements of verifyPinResp Object for success response:**

| S.No | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | Action Code **V** – Response Code **00**<br>Action Code **E** – Response Code **00** |
| 3 | respMsg | Response message | String | |
| 4 | actionCode | Validation or Execution | String | V or E |
| 5 | custCode | Customer Code | String | |

**Array Elements of verifyPinResp Object for failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | |
| 3 | respMsg | Response message | String | |
| 4 | actionCode | Validation or Execution | String | V or E |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/verifypin

Execution Request :

```
{
    “instCode” : “FIS”,
    “reqMsgId” : “201951968545010001”,
    “verifyPinReq”:[{
        “reqId” : “201908271919300000”,
        “actionCode”:”E”,
        “vpan” :”MYMC000000000739”,
         "pan":"XXXXXXXXXXXX",
      }]
}
```

Execution Response:

```
{
        “respCode”: “901”,
        “respMsg”: “Request Processed.”,
        “processCode”: 205,
        “instCode”: “FNCR”,
        “reqMsgId” : “201951968545010001”,
        “verifyPinResp”: {
        “verifyPinResp”: [
        {
            “reqId”: “201908271919300000”,
            “respCode”: “00”,
            “respMsg”: “ Pin Verified successfully.”,
            “actionCode”: “E”,
            “custCode”: “Cust00002”
        }]
       }
}
```

## 20. Transaction details Enquiry

Transaction enquiry used to get all transaction details of the account. We ca search / enquired based on below rules.

1. clearPan + startDate + endDate
2. vpan + startDate + endDate
3. accNo + startDate + endDate
4. rrNo

**Request Format:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | pFlag | 1 | M | A | S | If ‘S’, system will pocess synchronously else Deafult set ‘S’ [It always synchronous service] |
| 3 | reqId | 18 | M | N | Id of each request | |
| 4 | vpan | 15 - 19 | C | AN | Virtual Pan / Pan Alias | |
| 5 | clearPan | 19 | C | AN | Clear Card Number | |
| 6 | accNo | 0 - 28 | C | AN | Account Number | |
| 7 | rrNo | 12 | C | AN | Retrieval Reference Number | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | startDate | 8 | C | AN | Start Date of Transaction Date | Format is like **YYYYMMDD** |
| 9 | endDate | 8 | C | AN | End Date of Transaction Date | Format is like **YYYYMMDD** |
| 10 | reqMsgId | 18 | M | String | Request Message Id | |
| 11 | startRecordNo | 2 | M | N | Start Record Number | |
| 12 | endRecordNo | 2 | M | N | End Record Number | |

**Response Format:**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | instCode | Institution Code | String | |
| 2 | transactionDetailsRes | List of response | Array | Array of Object |
| 3 | processCode | Process id of each request | Number | |
| 4 | reqId | Request id of each request | AN | |
| 5 | reqMsgId | Request Message Id | String | |

**Elements of transactionDetailsRes Object for success response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | respCode | Response Code | String | |
| 2 | respMsg | Response Message | String | |
| 3 | transactionDetailsResp | List of response | Array | Array of Object |
| 4 | totalRecord | Overa all total No. of record | String | |

**Array Elements of transactionDetailsResp Object for success response:**

| SNo | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | tlogId | Transaction Id | N | |
| 2 | ctxDate | Cortex Business Date | N | Format should like YYYYMMDD |
| 3 | dateLocal | Transaction Local Date | N | Format should like YYYYMMDD |
| 4 | timeLocal | Transaction Local Time | N | |
| 5 | orgTxnDate | Original Transaction Date | N | Format should like YYYYMMDD |
| 6 | orgTxnTime | Orignal Transaction Time | N | |
| 7 | txnAmount | Transaction Amount | N | Transaction Amount with Decimals |
| 8 | billAmount | Billing Amount | N | Bill Amount with Decimals |

| | | | | |
|---|---|---|---|---|
| 9 | billCommission | Bill Commission Amount | N | Bill Commission Amount with Decimals [Issuer Commission] |
| 10 | txnFee | Transaction Fee | N | |
| 11 | txnCommission | Transaction Commission | AN | |
| 12 | txnRefNo | Transaction Referance Number | AN | |
| 13 | acqRefNo | Acquirer Referance Number | AN | |
| 14 | rrNo | Retrieval Referance Number | AN | |
| 15 | aiid | Acqurier Id | AN | |
| 16 | crdAccptId | Card Acceptor ID | AN | |
| 17 | crdAccptName | Card Acceptor Name | AN | |
| 18 | crdAccptStreet | Card Acceptor Street | AN | |
| 19 | crdAccptCity | Card Acceptor City | AN | |
| 20 | crdAccptPostCode | Card Acceptor Postal Code | AN | |
| 21 | crdAccptRegion | Card Acceptor Region | AN | |
| 22 | crdAccptCountry | Card Acceptor Country | AN | |
| 23 | terminalType | Terminl Type | AN | Ex: Action Code – Terminal Descr |
| 24 | terminalSeq | Terminal Sequence | N | Transaction Sequence Number from Terminal |
| 25 | transActionCode | Trasnaction Action Code | AN | Ex: Action Code – Action Code Descr |
| 26 | transResCode | Trasnaction Response Code | AN | |
| 27 | trasnRejReason | Transaction Reject Reason | N | |
| 28 | transVerifyRslt | Transaction Verification Result | AN | |
| 29 | transCode | Trasnaction Code | N | Ex: Transaction Code – Description |
| 30 | transDesc | Transaction Description | AN | |
| 31 | transStatus | Transaction Status | AN | Transaction Status<br>Ex: Code – Transation status<br>This status is different from the transaction status. Using this, we can identify whether the transaction is settled or reversed. Etc |
| 32 | pan | Encrypted Pan | AN | |
| 33 | panDisplay | Pan Display | AN | |
| 34 | poscdim | POSCDIM | AN | Card Data Input Method, Point of Sale<br>Value - Description |
| 35 | fiid | Forwarding Institution ID | AN | |
| 36 | fiidcountry | Forwarding Institution Country | AN | |
| 37 | billingCurr | Billing Currency | AN | |
| 38 | stan | STAN | N | |
| 39 | crdAccptLocation | Card Acceptor Location | AN | |
| 40 | terminalCode | Terminal Code | AN | |
| 41 | approvalCode | Approval Code | N | |
| 42 | accItemsId | Account Items Id | N | |
| 43 | feeid | Fee Id | N | |

| | | | | |
|---|---|---|---|---|
| 44 | revFlag | Reversal Flag | AN | |
| 45 | description | Transaction Description | AN | |
| 46 | extCode | Ext. Code | AN | |
| 47 | txnType | Transaction Type | N | |
| 48 | txnSubType | Transaction Sub Type | N | |
| 49 | loadType | Load Type | N | |
| 50 | loadSource | Load Source | N | |
| 51 | blockAmount | Blocked Amount with decimals | N | |
| 52 | availBalance | Available Balance | N | |
| 53 | accNo | Account Number | AN | |
| 54 | transactionCurrency | Transaction Currency | AN | |
| 55 | originalSTAN | Original STAN | AN | Original STAN (System Trace Audit Number) Present during Reversal Transactions |
| 56 | originaltransAmount | Original Transaction Amount | AN | |
| 57 | settlementDate | Settlement Date | AN | |
| 58 | settlementAmount | Settlement Amount | AN | |
| 59 | transmitDate | Transmit Date | AN | |
| 60 | transmitTime | Transmit Time | AN | |
| 61 | clearPan | Clear Pan | AN | Pan / Clear Pan value encrypted by Authman session key |

**Array Elements of transactionDetailsRes Object for failure response:**

| SNo | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | respCode | Response Code | String | |
| 2 | respMsg | Response message | String | |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/gettransactiondetails

**Sample Request:**

```
{
"instCode":"IEBT",
"reqMsgId" : "202012101049010002",
"pFlag":"s",
"reqId":"202012101049010002",
"accNo":"100000874001",
"startRecordNo":"1",
"endRecordNo":"10",
"startDate":"20201201",
"endDate":"20201210"
}
```

**Success Response:**

```
{
  "respCode": "901",
  "respMsg": "Request Processed.",
  "processCode": 211,
  "instCode": "IEBT",
  "reqMsgId": "202012101049010002",
  "reqId": "202012101049010002",
  "transactionDetailsRes": {
    "respCode": "00",
    "respMsg": "SUCCESS",
    "totalRecord": "2",
    "transactionDetailsResp": [
      {
        "tlogId": "6494",
        "ctxDate": "20201203",
        "dateLocal": "20201210",
        "timeLocal": "1934216",
        "orgTxnDate": "22630831",
        "orgTxnTime": "0",
        "txnAmount": "10000",
        "billAmount": "10000",
        "billCommission": "0",
        "txnFee": "0",
        "txnCommission": "0",
        "txnRefNo": " ",
        "acqRefNo": "438",
        "rrNo": "034514029034",
        "crdAccptId": "AGNT000000006",
        "crdAccptName": " ",
        "crdAccptStreet": " ",
        "crdAccptCity": " ",
        "crdAccptPostCode": " ",
        "crdAccptRegion": " ",
        "crdAccptCountry": "   ",
        "terminalType": "",
        "terminalSeq": "0",
        "transActionCode": "",
        "transResCode": "",
        "trasnRejReason": "",
        "transVerifyRslt": " ",
        "transCode": "201 - ACCOUNT LOAD",
        "transDesc": "ACCOUNT LOAD",
        "transStatus": "7 - Transaction complete,approved",
        "pan": "D6638C9AD13D3AD9B97",
        "panDisplay": "616161_1396",
        "poscdim": "  - ",
        "billingCurr": "356",
        "crdAccptLocation": " ",
        "terminalCode": " ",
        "approvalCode": " ",
        "accItemsId": "0",
        "revFlag": " ",
        "description": " ",
        "extCode": " ",
        "txnType": "-1",
        "txnSubType": "-1",
        "loadType": "-1",
        "loadSource": "-1",
        "blockAmount": "0",
        "availBalance": "0",
        "accNo": "100000874001",
        "transactionCurrency": "356",
        "originalSTAN": "0",
        "originaltransAmount": "0",
        "settlementDate": "22630831",
        "settlementAmount": "0",
        "transmitDate": "20201210",
        "transmitTime": "140427",
        "clearPan": " ",
        "aiid": " ",
        "fiid": " ",
        "fiidcountry": " ",
        "stan": "29034",
        "feeid": "0"
      },
      {
        "tlogId": "6493",
        "ctxDate": "20201203",
        "dateLocal": "20201210",
        "timeLocal": "1931533",
        "orgTxnDate": "22630831",
        "orgTxnTime": "0",
        "txnAmount": "101",
        "billAmount": "101",
        "billCommission": "0",
        "txnFee": "0",
        "txnCommission": "0",
        "txnRefNo": " ",
        "acqRefNo": "438",
        "rrNo": "034514030034",
        "crdAccptId": "AGNT000000005",
        "crdAccptName": " ",
        "crdAccptStreet": " ",
        "crdAccptCity": " ",
        "crdAccptPostCode": " ",
        "crdAccptRegion": " ",
        "crdAccptCountry": "   ",
        "terminalType": "",
        "terminalSeq": "0",
        "transActionCode": "",
        "transResCode": "",
        "trasnRejReason": "",
        "transVerifyRslt": " ",
        "transCode": "201 - ACCOUNT LOAD",
        "transDesc": "ACCOUNT LOAD",
        "transStatus": "7 - Transaction complete,approved",
        "pan": "D6638C9AD13D3AD9B97",
        "panDisplay": "616161_1396",
        "poscdim": "  - ",
        "billingCurr": "356",
        "crdAccptLocation": " ",
        "terminalCode": " ",
        "approvalCode": " ",
        "accItemsId": "0",
        "revFlag": " ",
        "description": " ",
        "extCode": " ",
        "txnType": "-1",
        "txnSubType": "-1",
        "loadType": "-1",
        "loadSource": "-1",
        "blockAmount": "0",
        "availBalance": "0",
        "accNo": "100000874001",
        "transactionCurrency": "356",
        "originalSTAN": "0",
        "originaltransAmount": "0",
        "settlementDate": "22630831",
        "settlementAmount": "0",
        "transmitDate": "20201210",
        "transmitTime": "140157",
        "clearPan": " ",
        "aiid": " ",
        "fiid": " ",
        "fiidcountry": " ",
        "stan": "30034",
        "feeid": "0"
      }
    ]
  }
}
```

**Failure Response:**

```
{
  "processCode": 211,
  "instCode": "FIS",
  "reqMsgId": "201951968545010001",
  "reqId": "201909160332160002",
  "transactionDetailsRes": {
    "respCode": "01",
    "respMsg": "NODATA"
  }
}
```

## 21. Gift Card Order

Gift Card Order API is used to place card order for gift cards. Only the request will be logged in the system. As part of EOD process, the requested cards will be generated.

**Request Format:**

| S.No. | Field | Min-Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 - 4 | M | AN | Institution Code | |
| 2 | giftCardOrderReq | 1-50 | M | Array | | Array of Object |
| 3 | pFlag | 0-1 | O | A | S | If S, system will pocess synchronously else async |
| 4 | reqMsgId | 18 - 18 | M | String | Request Message Id | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Array Elements of giftCardOrderReq Request Object:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | reqId | 18 -18 | M | N | Request id of each request | |
| 2 | actionCode | 1 -1 | M | A | Validation or Execution | V or E |
| 3 | branchCode | 1-8 | M | AN | Branch Code | |
| 4 | cardProduct | 1-4 | M | AN | Card Product | |
| 5 | cardCount | 1-5 | M | N | Card Count | |
| 6 | companyName | 0-50 | O | ANS | This name will be used as emboss name while card creation. | |
| 7 | User | 1-20 | M | ANS | User, who will place card Order | |
| 8 | persoBureauName | 0-20 | O | ANS | | Emboss Vendor Name |
| 9 | userData1 | 0-12 | O | ANS | | User Data 1 ( For Future ) |
| 10 | userData2 | 0-12 | O | ANS | | User Data 2 ( For Future ) |
| 11 | amountLoadFlag | 1 | M | A | Y/N | If amount to be loaded in gift card |
| 12 | loadAmount | 0-12 | CM | ANS | | Amount to be loaded in gift card. Mandatory if amountLoadFlag is Y. |

**Response Format:**

| S.No. | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | instCode | String | Institution Code | |
| 2 | giftCardOrderRes | JSON Object | JSON Object | |
| 3 | processCode | Number | Process Code | |
| 4 | reqMsgId | String | Request Message Id | |
| 5 | respCode | String | Response Code | |
| 6 | respMsg | String | Response message | |

**Elements of giftCardOrder Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | giftCardOrderResponseList | List of response | Array | Array of Object |

**Array Elements of giftCardOrderResponseList Response Object for success response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response Message | |
| 4 | actionCode | String | Validation or Execution | V or E |
| 5 | branchCode | String | Branch Code | |
| 6 | cardProduct | String | Card Product | |
| 7 | cardCount | String | Card Count | |
| 8 | invoiceNo | String | Random number generated for card order requet | It will be generated only for execution request not for validation. |
| 9 | User | String | User, who will place card Order | |

**Array Elements of giftCardOrderResponseList Response Object for failure response:**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | reqId | String | Request id of each request | |
| 2 | respCode | String | Response Code | |
| 3 | respMsg | String | Response message | |
| 4 | actionCode | String | Validation or Execution | V or E |

URL : http<s>://<ip>:<port>/app /rest/v1.0/giftcardorder

**Validation Request** :

```
{
“instCode” : “CLNT”,
“reqMsgId” : “201951968545010001”,
“giftCardOrderReq”:[
        {
        “reqId” : “201908271919300000”,
        “actionCode”:”V”,
        “branchCode” : “0001”,
        “cardProduct” : “MYMC”,
        “cardCount” : “00001”,
        “user” : “user1”,
        “amountLoadFlag”:”N”
        },
        {
        “reqId” : “201908271919300001”,
        “actionCode”:”V”,
        “branchCode” : “0001”,
        “cardProduct” : “MYMC”,
        “cardCount” : “00001”,
        “user” : “user2”,
        “amountLoadFlag”:”Y”,
        “loadAmount”:”20000”
        }]
}
```

**Validation Response :**

```
{
                 “respCode”: “901”,
                “respMsg”: “Request Processed.”,
                “processCode”: 101,
                “instCode”: “CLNT”,
               “reqMsgId” : “201951968545010001”,

    “giftCardOrderRes”: {

                “giftCardOrderResponseList”: [
                    {
                       “reqId”: “201908271919300000”,
                       “respCode”: “00”,
                       “respMsg”: “Gift Card Order Validation Done”,
                       “actionCode”: “V”,
                       “branchCode”: “0001”,
                       “cardProduct”: “MYMC”,
                       “cardCount”: “00001”,
                       “user”: “user1222222222222222”,
                       “amountLoadFlag”:”N”
                    },
                    {
                       “reqId”: “201908271919300001”,
                       “respCode”: “00”,
                       “respMsg”: “Non Perso Card Order Validation Done”,
                       “actionCode”: “V”,
                       “branchCode”: “0001”,
                       “cardProduct”: “MYMC”,
                       “cardCount”: “00001”,
                       “user”: “user3”,
                       “amountLoadFlag”:”Y”,
                        “loadAmount”:”20000”
                    }
```

```
          ]
        }
}
```

**Execution REQUEST**

```
        {
        “instCode” : “CLNT”,
        “pFlag” : “”,
        “reqMsgId” : “201951968545010001”,

        “giftCardOrderReq”:[
        {
        “reqId” : “201908271919300000”,
        “actionCode”:”E”,
        “branchCode” : “0001”,
        “cardProduct” : “MYMC”,
        “cardCount” : “00001”,
        “user” : “user1”,
        “amountLoadFlag”:”N”

        },
        {
        “reqId” : “201908271919300001”,
        “actionCode”:”E”,
        “branchCode” : “0001”,
        “cardProduct” : “MYMC”,
        “cardCount” : “00001”,
        “user” : “user2” ,
                “amountLoadFlag”:”Y”,
                 “loadAmount”:”20000”
        }]
}
```

**SUCCESS RESPONSE**

```
        {
          “respCode”: “901”,
          “respMsg”: “Request Processed.”,
           “processCode”: 101,
           “instCode”: “CLNT”,
       “reqMsgId” : “201951968545010001”,

  “giftCardOrderRes”: {

           “giftCardOrderResponseList”: [
             {
                “reqId”: “201908271919300000”,
                “respCode”: “00”,
                “respMsg”: “Bulkiss record processed Successfully”,
                “actionCode”: “E”,
                “branchCode”: “0001”,
                “cardProduct”: “MYMC”,
                “cardCount”: “00001”,
                “invoiceNo”: “000000000446”,
                “user”: “user1” ,
                “amountLoadFlag”:”N”
             },
             {
                “reqId”: “201908271919300001”,
                “respCode”: “00”,
                “respMsg”: “Bulkiss record processed Successfully”,
                “actionCode”: “E”,
                “branchCode”: “0001”,
                “cardProduct”: “MYMC”,
                “cardCount”: “00001”,
                “invoiceNo”: “000000000447”,
                “user”: “user2” ,
                “amountLoadFlag”:”Y”,
                 “loadAmount”:”20000”
             }   ] }
     }
```

**FAILURE RESPONSE**

```
{
          “respCode”: “901”,
         “respMsg”: “Request Processed.”,
          “processCode”: 101,
          “instCode”: “CLNT”,
         “giftCardOrderRes”: {
          “giftCardOrderResponseList”: [
            {
              “reqId”: “201908271919300000”,
              “respCode”: “01”,
              “respMsg”: “Branch [00000001] not found”,
               “actionCode”:”E”

            },
            {
              “reqId”: “201908271919300001”,
              “respCode”: “01”,
              “respMsg”: “Branch [0002] not found”,
              “actionCode”:”E”
            }
          ]}
}
```

## 22. CVV2 Verification

CVV2 Verification API is to be verify the CVV2 value for a card. CVV2 value shall be sent in the request.

**Request Format**:

| S. No | Field | Min-Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institution Code | Example FIS |
| 2 | reqId | 18 | M | N | Id of each request, uniquely identified | |
| 3 | pan | 16-19 | C | Character | Clear card number (or number used for PIN block formation) | |
| 4 | vpan | 0-32 | C | Character | Virtual PAN (or) PAN Alias | |
| 5 | cvv2 | 3 | O | String | | |
| 6 | expDate | 1 | M | Number | Expiry date | Expiry date in YYMM format. |

**Response Format**:

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | instCode | Institution Code | String | |
| 2 | reqMsgId | Request id of each request | String | |
| 3 | respCode | Response Code | String | |
| 4 | respMsg | Response message | String | |
| 5 | **verifyCvvRes** | JSON Object | JSON Object | |
| 6 | processCode | Process code | | |

**Elements of verifyCvvRes Object for response**

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | **verifyCvvResp** | List of response | Array | Array of Object |

**Array Elements of verifyCvvResp Object for success response:**

| S.No | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | Action Code **V** – Response Code **00**<br>Action Code **E** – Response Code **00** |
| 3 | respMsg | Response message | String | |
| 4 | vpan | Vpan | String | Based on req it will pe pan or vpan |
| 5 | mobileNumber | Mobile Number | String | |

| | | | | |
|---|---|---|---|---|
| 6 | firstName | First Name | String | |
| 7 | accNo | Account Number | String | |
| 8 | custCode | Customer Code | String | |
| 9 | cvv2 | Cvv2 value | String | |
| 10 | expDate | Expiry Date | String | |
| | | | | |

**Failure response:**

| **Sno** | **Field** | **Description** | **DataType** | **Comments** |
|---|---|---|---|---|
| 1 | reqId | Request id of each request | String | |
| 2 | respCode | Response Code | String | |
| 3 | respMsg | Response message | String | |
| 4 | instCode | Institution Code | String | |

**Note: Either PAN or VPAN should be given as input.**

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/verifycvv

CVV2 Verification Request:

```
{
        "instCode":"IEBT",
        "reqMsgId":"202011251341301253",
        "pFlag":"s",
        "verifyCvvReq":[
                {
                        "reqId" : "202011251341301253",
                        "pan":"6161610008671396",
                        "cvv2":"137",
                        "expDate":"2506",
                        "mobileNumber":"91996200478",
                        "firstName":"Prabha",
                        "accNo":"0000400046456",
                        "custCode":"CUST00023"
                }
        ]
}
```

CVV2 Verifcation Response:

```
{
   "respCode": "901",
   "respMsg": "Request Processed.",
   "processCode": 227,
   "instCode": "IEBT",
   "reqMsgId": "202011251341301253",
   "verifyCvvRes": {
      "verifyCvvResp": [
         {
            "respMsg": "CVV verification successful",
            "respCode": "00",
            "cvv2": "137",
            "expDate": "2506"
         }
      ]
   }
}
```

Failure Response :

```
{
   “instCode”: “FIS”,
   “reqId”: “201908271719300000”,
   “respCode”: “01”,
   “respMsg”: “CVV verification failed”
}
```

## 23. Card Details Enquiry

Card details enquiry to get the card product limits and card limits along with channel stauts whether the customer has enabled or disabled. It allows both pan abd VPAN.

**Request Format**:

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | pFlag | 1 | O | A | S | If ‘S’, system will pocess synchronously else Deafult set ‘S’ [It always synchronous service] |
| 3 | reqMsgId | 18 | M | N | Id of each request | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | pan | 19 | C | AN | Clear Card Number. | clearPan |
| 5 | vpan | 15 - 19 | C | AN | Virtual Card Number / PAN ALIAS | vpan |

**Response Format**:

| S.No. | Field | Description | DataType | Comments |
|---|---|---|---|---|
| 1 | instCode | Institution Code | AN | |
| 2 | limitEnquiryRes | Response Object | Object | Json Object |
| 3 | reqMsgId | Request id of each request | AN | |
| 4 | processCode | Number | Process id of each request | |
| 5 | reqMsgId | String | Request Message Id | |

**Elements of limitEnquiryRes Object for success response**

| Sno | Field | Data Type | Description | Comments |
|---|---|---|---|---|
| 1 | custCode | String | Customer Code | |
| 2 | pan | String | Card Number | Clear or Encrypted Pan |
| 3 | vpan | String | Card Alias | |
| 3 | cardStatus | String | Card Status | |
| 4 | cardExpiryDate | String | Expiry date | |
| 5 | cardProduct | String | Card Product | |
| 6 | domAtm | String | | Allowed Values are 'Y' or 'N' |
| 7 | domPos | String | | Allowed Values are 'Y' or 'N' |
| 8 | domEcom | String | | Allowed Values are 'Y' or 'N' |
| 9 | domConless | String | | Allowed Values are 'Y' or 'N' |
| 10 | intlAtm | String | | Allowed Values are 'Y' or 'N' |
| 11 | intlPos | String | | Allowed Values are 'Y' or 'N' |
| 12 | intlEcom | String | | Allowed Values are 'Y' or 'N' |
| 13 | intlConless | String | | Allowed Values are 'Y' or 'N' |
| 14 | linkedLimits | Json Object | | |
| 15 | cardProductLimits | Json Object | | |

**Json object of linkedLimits object:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | domAtmLimit | | C | String | | Domestic ATM Amount Limit to be update |
| 2 | domPosLimit | | C | String | | Domestic POS Amount Limit to be update |

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | domEcomLimit | | C | String | | Domestic ECOM Amount Limit to be update |
| 4 | domConlessLimit | | C | String | | Domestic Contact less Amount Limit to be update |
| 5 | intlAtmLimit | | C | String | | International ATM Amount Limit to be updates |
| 6 | intlPosLimit | | C | String | | International POS Amount Limit to be update |
| 7 | intlEcomLimit | | C | String | | International ECOM Amount Limit to be update |
| 8 | intlConlessLimit | | C | String | | International Contact less Amount Limit to be update |

**Json object of cardProductLimits object:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | domAtm | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 2 | domPos | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 3 | domEcom | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 4 | domConless | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 5 | intlAtm | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 6 | intlPos | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 7 | intlEcom | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 8 | intlConless | | C | String | | |

**Array Elements of limitEnquiryRes Object for failure response:**

| Sno | Field | Description | DataType | Comments |
|---|---|---|---|---|
| **1** | reqMsgId | Request id of each request | String | |
| 2 | respCode | Response Code | String | **99** |
| 3 | respMsg | Response message | String | **No Data** |

**URL :** **http<s>://<ip>:<port>/app/rest/v1.0/carddetails**

**Sample Request:**

```
{
        "instCode":"IEBT",
        "reqMsgId":"202011171711301250",
        "pFlag":"s",
        "pan":"6161610008671628"
}
```

**Sample Response:**

```
{
  "respCode": "901",
  "respMsg": "Request Processed.",
  "processCode": 224,
  "instCode": "EVRG",
  "reqMsgId": "202011171711301250",
  "reqId": "202011171711301250",
  "cardFetchRes": {
    "limitEnquiryResp": [
      {
        "domAtm": "Y",
        "domPos": "Y",
        "domEcom": "Y",
        "domConless": "Y",
        "intlAtm": "N",
        "intlPos": "N",
        "intlEcom": "N",
        "intlConless": "N",
        "linkedLimits": {
          "domAtmLimit": "4000.00",
          "domPosLimit": "5000.00",
          "domEcomLimit": "1000.00",
          "domConlessLimit": "200.00",
          "intlAtmLimit": "3000.00",
          "intlPosLimit": "2000.00",
          "intlEcomLimit": "4000.00",
          "intlConlessLimit": "5000.00"
        },
        "cardProductLimits": {
          "domAtmLimit": "10000.00",
          "domPosLimit": "10001.00",
          "domEcomLimit": "10002.00",
          "domConlessLimit": "10003.00",
          "intlAtmLimit": "20000.00",
          "intlPosLimit": "20001.00",
          "intlEcomLimit": "20002.00",
          "intlConlessLimit": "20003.00"
        }
      }
    ]
  }
```

}**Failure Response:**

```
 {
          “processCode”: 224,
          “instCode”: “CLNT”,
          “reqMsgId”: “201951968545010001”,
          “reqId”: “201908271919300000”,
          “limitEnquiryRes”: {
            “reqId”: “201908271919300000”,
            “respCode”: “01”,
            “respMsg”: “NODATA”
          }
        }
```

## 24. Channel Update

Channel update helps to update the channels either to allow or deny during transactions. It allows to update all the channels like ATM, POS, ECOM and Contact less for both domestic and international.

**Request Format**:

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | pFlag | 1 | O | A | S | If 'S', system will pocess synchronously else Deafult set 'S' [It always synchronous service] |
| 3 | reqMsgId | 18 | M | N | Id of each request | |
| 4 | pan | 19 | C | AN | Clear Card Number. | clearPan |
| 5 | vpan | 15 - 19 | C | AN | Virtual Card Number / PAN ALIAS | vpan |
| 6 | domAtm | 0-1 | C | String | | Allowed Values are 'Y' or 'N' |
| 7 | domPos | 0-1 | C | String | | Allowed Values are 'Y' or 'N' |
| 8 | domEcom | 0-1 | C | String | | Allowed Values are 'Y' or 'N' |
| 9 | domConless | 0-1 | C | String | | Allowed Values are 'Y' or 'N' |
| 10 | intlAtm | 0-1 | C | String | | Allowed Values are 'Y' or 'N' |
| 11 | intlPos | 0-1 | C | String | | Allowed Values are 'Y' or 'N' |
| 12 | intlEcom | 0-1 | C | String | | Allowed Values are 'Y' or 'N' |
| 13 | intlConless | 0-1 | C | String | | Allowed Values are 'Y' or 'N' |
| 14 | makerId | 0-50 | O | String | Maker User Details | |
| 15 | checkerId | 0-50 | O | String | Checker User Details | |
| 16 | approverId | 0-50 | O | String | Approver Details | |

**Response Format:**

| S.No. | Field | Description | Comments |
|---|---|---|---|
| 1 | instCode | Institution Code | |
| 2 | channelUpdateRes | JSON Object | JSON Object |
| 3 | processCode | Number | Process id of each request |
| 4 | reqMsgId | String | Request Message Id |
| 5 | respCode | String | Response Code |
| 6 | respMsg | String | Response message |

**Elements of channelUpdateRes Object for response**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | responseCode | 2-3 | M | String | | Response code of the request |
| 2 | status | 1-50 | M | String | | Response Description |

**URL :** **http<s>://<ip>:<port>/app/rest/v1.0/channelupdate**

**Sample Request**:

```
{
"instCode":"FIS",
"reqMsgId":"201908271919300000",
"clearPan":"dER1W898jiui2o45zmgMoA==. Y2dJsYeA1cuYiF66e4osVA==",
"vpan":"PMVC000000000299",
"makerId":"",
"checkerId":"",
"approverId":"",
"domAtm":"Y",
"domPos":"Y",
"domEcom":"Y",
"domConless":"Y",
"intlAtm":"Y",
"intlPos":"Y",
"intlEcom":"Y",
"intlConless":"Y"
}
```

**Success Response**:

```
{
"instCode":"FIS",
"reqMsgId":"201908271919300000",
"processCode":"223",
"respCode":"901",
"respMsg":"Request Processed.",
"channelUpdateRes": {
"responseCode":"000",
"status":"Success"
}
}
```

**Failure Response**:

```
{
"respCode": "904",
"respMsg": "Missing the pan / vPan",
"processCode": 223,
"instCode": "FIS",
"reqMsgId": "202051968545010005",
"reqId": "202008271919300004"
}
```

## 25. Limit Update

API used to update card limits for both Domestic and International. It allows both PAN and VPAN.

**Request Format**:

| **Sno** | **Field** | **Max Length** | **Mandatory?** | **Data Type** | **Description** | **Comments** |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | pFlag | 1 | O | A | S | If ‘S’, system will pocess synchronously else Deafult set ‘S’ [It always synchronous service] |
| 3 | reqMsgId | 18 | M | N | Id of each request | |
| 4 | pan | 19 | C | AN | Clear Card Number. | clearPan |
| 5 | vpan | 15 - 19 | C | AN | Virtual Card Number / PAN ALIAS | vpan |
| 4 | domAtmLimit | | C | String | | Domestic ATM Amount Limit to be update |
| 5 | domPosLimit | | C | String | | Domestic POS Amount Limit to be update |
| 6 | domEcomLimit | | C | String | | Domestic ECOM Amount Limit to be update |

| | | | | | | |
|---|---|---|---|---|---|---|
| 7 | domConlessLimit | | C | String | | Domestic Contact less Amount Limit to be update |
| 8 | intlAtmLimit | | C | String | | International ATM Amount Limit to be updates |
| 9 | intlPosLimit | | C | String | | International POS Amount Limit to be update |
| 10 | intlEcomLimit | | C | String | | International ECOM Amount Limit to be update |
| 11 | intlConlessLimit | | C | String | | International Contact less Amount Limit to be update |
| 12 | makerId | 0-50 | O | String | Maker User Details | |
| 13 | checkerId | 0-50 | O | String | Checker User Details | |
| 14 | approverId | 0-50 | O | String | Approver Details | |

**Response Format**:

| S.No. | Field | Description | Comments |
|---|---|---|---|
| 1 | instCode | Institution Code | |
| 2 | limitUpdateRes | JSON Object | JSON Object |
| 3 | processCode | Number | Process id of each request |
| 4 | reqMsgId | String | Request Message Id |
| 5 | respCode | String | Response Code |
| 6 | respMsg | String | Response message |

**Elements of limitUpdateRes Object for response**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | responseCode | 2-3 | M | String | | Response code of the request |
| 2 | status | 1-50 | M | String | | Response Description |

**URL : http<s>://<ip>:<port>/app/rest/v1.0/cardlimit**

**Sample Request**:

```
{
"instCode":"FIS",
"reqMsgId ":"201908271919300000",
"clearPan":"dER1W898jiui2o45zmgMoA==. Y2dJsYeA1cuYiF66e4osVA==",
"vpan":"PMVC000000000299",
        "domAtmLimit":"1000",
        "domPosLimit":"1000",
        "domEcomLimit":"1000",
        "domConlessLimit":"1000",
        "intlAtmLimit":"1000",
        "intlPosLimit":"1000",
        "intlEcomLimit":"1000",
        "intlConlessLimit":"1000"
}
```

**Sample Response**:

```
{
"instCode":"FIS",
"reqMsgId":"201908271919300000",
"processCode":"225",
"respCode":"901",
"respMsg":"Request Processed ",
"limitUpdateRes":{
"responseCode":"000",
"status":"Success"
}
}
```

**Failure Response**:

```
{
"respCode": "904",
"respMsg": "Missing the pan / vPan",
"processCode": 225,
"instCode": "FIS",
"reqMsgId": "202051968545010005",
"reqId": "202008271919300004"
}
```

## 26. Card Limit and Channel Update

Card Limit and Channel update helps to update both limit and channel. Here also VPAN and PAN allowed

**Request Format**:

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | instCode | 4 | M | AN | Institute Code | |
| 2 | pFlag | 1 | O | A | S | If ‘S’, system will pocess synchronously else Deafult |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | set ‘S’ [It always synchronous service] |
| 3 | reqMsgId | 18 | M | N | Id of each request | |
| 4 | pan | 19 | C | AN | Clear Card Number. | clearPan |
| 5 | vpan | 15 - 19 | C | AN | Virtual Card Number / PAN ALIAS | vpan |
| 6 | limitDetails | 1-1 | M | Json Object | Limit details | |
| 7 | channelAccessDetails | 1-1 | M | Json Object | Channel Access Details | |
| 5 | makerId | 0-50 | O | String | Maker User Details | |
| 6 | checkerId | 0-50 | O | String | Checker User Details | |
| 7 | approverId | 0-50 | O | String | Approver Details | |

**Elements of limitDetails object:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | domAtmLimit | | C | String | | Domestic ATM Amount Limit to be update |
| 2 | domPosLimit | | C | String | | Domestic POS Amount Limit to be update |
| 3 | domEcomLimit | | C | String | | Domestic ECOM Amount Limit to be update |
| 4 | domConlessLimit | | C | String | | Domestic Contact less Amount Limit to be update |
| 5 | intlAtmLimit | | C | String | | International ATM Amount Limit to be updates |
| 6 | intlPosLimit | | C | String | | International POS Amount Limit to be update |
| 7 | intlEcomLimit | | C | String | | International ECOM Amount Limit to be update |
| 8 | intlConlessLimit | | C | String | | International Contact less Amount Limit to be update |

**Elements of channelAccessDetails object:**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | domAtm | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 2 | domPos | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 3 | domEcom | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 4 | domConless | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 5 | intlAtm | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 6 | intlPos | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 7 | intlEcom | | C | String | | Allowed Values are ‘Y’ or ‘N’ |
| 8 | intlConless | | C | String | | |

**NOTE:**

- To update card limits at least one Limit Field is Mandatory.
- To Update Card channel access at least One Field is Mandatory.

**Response Format**:

| S.No. | Field | Description | Comments |
|---|---|---|---|
| 1 | instCode | Institution Code | |
| 2 | limitAndChannelUpdateRes | JSON Object | JSON Object |
| 3 | processCode | Number | Process id of each request |
| 4 | reqMsgId | String | Request Message Id |
| 5 | respCode | String | Response Code |
| 6 | respMsg | String | Response message |

**Elements of limitAndChannelUpdateRes Object for response**

| Sno | Field | Max Length | Mandatory? | Data Type | Description | Comments |
|---|---|---|---|---|---|---|
| 1 | responseCode | 2-3 | M | String | | |
| 2 | status | 1-50 | M | String | | Response Description |

**URL :** http<s>://<ip>:<port>/app/rest/v1.0/ cardlimitandchannel

**Sapmle Request**:

```
{
"instCode":"FIS",
“reqMsgId” : “201951968545010001”,
"pan":"dER1W898jiui2o45zmgMoA==. Y2dJsYeA1cuYiF66e4osVA==",
“vpan”:”PMVC000000000299”,
"makerId":"",
"checkerId":"",
"approverId":"",
"limitDetails":{
"domAtmLimit":"1000",
"domPosLimit":"1000",
"domEcomLimit":"1000",
"domConlessLimit":"1000",
"intlAtmLimit":"1000",
"intlPosLimit":"1000",
"intlEcomLimit":"1000",
"intlConlessLimit":"1000"
},
"channelAccessDetails":{
"domAtm":"Y",
"domPos":"Y",
"domEcom":"Y",
"domConless":"Y",
"intlAtm":"Y",
"intlPos":"Y",
"intlEcom":"Y",
"intlConless":"Y"
}
}
```

**Response Format:**

```
{
"instCode":"FIS",
"reqMsgId":"201908271919300000",
"processCode":"223",
"respCode":"901",
"respMsg":"Request Processed.",
"limitAndChannelUpdateRes": {
"responseCode":"000",
"status":"Success"
}
}
```

## 27. ERROR CODE AND MESSAGE

| Code | Message |
|---|---|
| 196 | Failure, need to send request again |
| 401 | Unauthorized, Request declined. Need proper authentication to process the request. |
| 202 | Success, Request Accepted and pushed to the process queue |
| 901 | Request Processed. |
| 902 | No Records Found |
| 903 | Request Id (reqid) Not Found or Not Valid |
| 904 | Someting went wrong |
| 905 | Unable to process more than 6 requests |

| | |
|---|---|
| 906 | Application Error |
| 907 | Applicaion down. Try again later."; |
| 908 | Inst Code mismatch |
| 909 | Timed Out |

**To process Async request, below response code will be used for acknowledgement. Each request will be acknowledged with one of these codes.**

| Code | Message |
|---|---|
| 202 | Success, Request Accepted and pushed to the process queue |
| 401 | Unauthorized, Request declined. Need proper authentication to process the request. |
| 196 | Failure, need to send request again |
| 907 | Applicaion down. Try again later."; |

**Source should retry the below response codes.**

| Code | Message |
|---|---|
| 196 | Failure, need to send request again |
| 401 | Unauthorized, Request declined. Need proper authentication to process the request. |
| 904 | Cortex Down. Try later |
| 905 | Unable to process more than 6 request |
| 906 | Application Error |
| 907 | Applicaion down. Try again later."; |
| 909 | Timed Out |

# Annxure 1 – Authman integration – java code for reference

```
package client;
import java.security.Key;
import java.security.SecureRandom;
import java.security.Security;
import java.util.Base64;

import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

import org.bouncycastle.jce.provider.BouncyCastleProvider;
```

```
public class EncryptFields {
        public static void main(String[] args) throws Exception {
                String base64SessionKey = "c2liMTIzNDU2Nzg5MTIzNA==";
                String instId = "12345678";
                Security.addProvider(new BouncyCastleProvider());
                EncryptFields enc = new EncryptFields();
                System.out.println(enc.encrypt(instId, base64SessionKey));
        }

        public String encrypt(final String plainValue, final String key) throws Exception {
                Cipher cipher = this.getEncryptionCipher(key);
                final byte[] encryptedBytes = cipher.doFinal(plainValue.getBytes("UTF-8"));
                byte[] iV = cipher.getIV();
                return new String(Base64.getEncoder().encode(encryptedBytes)) + "."
                                + new String(Base64.getEncoder().encode(iV));
        }

        private Cipher getEncryptionCipher(final String key) throws Exception {
                final Key aesKey = new SecretKeySpec(Base64.getDecoder().decode(key), "AES");
                final Cipher encryptionCipher = Cipher.getInstance("AES/CBC/PKCS5Padding",
"BC");
                IvParameterSpec ivParameterSpec = new IvParameterSpec(generateIv());
                encryptionCipher.init(Cipher.ENCRYPT_MODE, aesKey, ivParameterSpec);
                return encryptionCipher;
        }

        private byte[] generateIv() {
                SecureRandom random = new SecureRandom();
                byte[] ivBytes = new byte[16];
                random.nextBytes(ivBytes);
                return ivBytes;
        }

}
```