API INTERACTION PROTOCOL

API developer reference

March 2017

Contents

[1 PREFACE 4](#_Toc514250019)

[1.1 Revision history 4](#_Toc514250020)

[1.2 Document purpose 5](#_Toc514250021)

[2 SMARTVISTA API INTERACTION OVERVIEW 6](#_Toc514250022)

[2.1 General concepts 6](#_Toc514250023)

[3 CUSTOMERS 6](#_Toc514250024)

[3.1 Overview 6](#_Toc514250025)

[3.2 References 6](#_Toc514250026)

[3.3 List of PROCEDURES 6](#_Toc514250027)

[3.3.1 GET\_CUSTOMER\_LIST 6](#_Toc514250028)

[3.3.2 GET\_FIRST\_CUSTOMER\_ID 7](#_Toc514250029)

[3.3.3 GET\_MAIN\_CUSTOMER\_INFO 8](#_Toc514250030)

[3.3.4 GET\_CUSTOMER\_LIMIT\_INFO 11](#_Toc514250031)

[3.3.5 GET\_CONTACTS\_CUSTOMER\_INFO 12](#_Toc514250032)

[3.3.6 GET\_ADDRESSES\_CUSTOMER\_INFO 13](#_Toc514250033)

[4 ACCOUNTS 14](#_Toc514250034)

[4.1 OVERVIEW 14](#_Toc514250035)

[4.2 REFERENCES 14](#_Toc514250036)

[4.3 LIST OF PROCEDURES 14](#_Toc514250037)

[4.3.1 GET\_MAIN\_BILLING\_INFO 15](#_Toc514250038)

[5 TRANSACTIONS 16](#_Toc514250039)

[5.1 OVERVIEW 16](#_Toc514250040)

[5.2 REFERENCES 16](#_Toc514250041)

[5.3 LIST OF PROCEDURES 16](#_Toc514250042)

[5.3.1 GET\_TRANSACTIONS\_DATA 16](#_Toc514250043)

[5.3.2 GET\_GL\_ACCOUNT\_NUMBERS\_DATA 19](#_Toc514250044)

[5.3.3 GET\_ACTIVE\_ACCOUNTS\_FOR\_PERIOD 19](#_Toc514250045)

[5.3.4 GET\_LINK\_ACCOUNT\_BALANCES 20](#_Toc514250046)

[6 CARD 21](#_Toc514250047)

[6.1 OVERVIEW 21](#_Toc514250048)

[6.2 REFERENCES 21](#_Toc514250049)

[6.3 LIST OF PROCEDURES 21](#_Toc514250050)

[6.3.1 GET\_MAIN\_CARDS\_INFO 22](#_Toc514250051)

[6.3.2 GET\_CUSTOMER\_MARKETING\_INFO 24](#_Toc514250052)

[6.3.3 CHANGE\_DELIVERY\_STATUS\_DATE 25](#_Toc514250053)

[6.3.4 GET\_CARD\_TYPE\_FEATURE 25](#_Toc514250054)

[7 NAMING FORMAT 26](#_Toc514250055)

[7.1 OVERVIEW 26](#_Toc514250056)

[7.2 REFERENCES 26](#_Toc514250057)

[7.3 LIST OF PROCEDURES 26](#_Toc514250058)

[7.3.1 IMPORT\_POOL\_VALUE 26](#_Toc514250059)

[7.3.2 IMPORT\_POOL\_VALUES 26](#_Toc514250060)

[8 CREDITS 27](#_Toc514250061)

[8.1 Overview 27](#_Toc514250062)

[8.2 References 27](#_Toc514250063)

[8.3 List of PROCEDURES 27](#_Toc514250064)

[8.3.1 ACCOUNT\_STATEMENT 27](#_Toc514250065)

[8.3.2 LOYALTY\_POINTS\_SUM 28](#_Toc514250066)

[8.3.3 CREDIT\_SERVICE\_LIMIT 29](#_Toc514250067)

[8.3.4 ACCOUNT\_CARD 30](#_Toc514250068)

[8.3.5 DUE\_AMOUNTS 30](#_Toc514250069)

[8.3.6 DUE\_AMOUNTS 31](#_Toc514250070)

[8.3.7 GET\_CUSTOMER\_CRD\_INVOICE\_INFO 32](#_Toc514250071)

[8.3.8 GET\_CRD\_INVOICE\_PAYMENTS\_INFO 33](#_Toc514250072)

[9 Visa 33](#_Toc514250073)

[9.1 Overview 34](#_Toc514250074)

[9.2 References 34](#_Toc514250075)

[9.3 List of PROCEDURES 34](#_Toc514250076)

[9.3.1 GET\_UNHELD\_TRANSACTIONS 34](#_Toc514250077)

[9.3.2 GET\_DEDUCTED\_TRANSACTIONS 35](#_Toc514250078)

[9.3.3 GET\_TRANSACTIONS\_WITHOUT\_STTL 36](#_Toc514250079)

[9.3.4 GET\_TRANSACTIONS\_WITHOUT\_AUTH 37](#_Toc514250080)

[10 Scoring 38](#_Toc514250081)

[10.1 Overview 38](#_Toc514250082)

[10.2 Referencies 38](#_Toc514250083)

[10.3 List of PROCEDURES 38](#_Toc514250084)

[10.3.1 ADD\_BUCKET 38](#_Toc514250085)

[10.3.2 GET\_SCORING\_DATA 38](#_Toc514250086)

[11 Events 41](#_Toc514250087)

[11.1 Overview 41](#_Toc514250088)

[11.2 References 41](#_Toc514250089)

[11.3 List of procedures 41](#_Toc514250090)

[11.3.1 GET\_EVENTS 41](#_Toc514250091)

[12 OPERATIONS 42](#_Toc514250092)

[12.1 OVERVIEW 42](#_Toc514250093)

[12.2 REFERENCES 42](#_Toc514250094)

[12.3 LIST OF PROCEDURES 42](#_Toc514250095)

[12.3.1 GET\_OPERATIONS\_DATA 42](#_Toc514250096)

[12.3.2 GET\_OPER\_PARTICIPANTS\_DATA 46](#_Toc514250097)

[12.3.3 GET\_AGGR\_OPER\_TRANSACT\_DATA 47](#_Toc514250098)

[13 Payment orders 48](#_Toc514250099)

[13.1 OVERVIEW 48](#_Toc514250100)

[13.2 REFERENCES 48](#_Toc514250101)

[13.3 LIST OF PROCEDURES 48](#_Toc514250102)

[13.3.1 GET\_ORDER\_LIST 49](#_Toc514250103)

[13.3.2 GET\_ORDER\_LIST\_COUNT 49](#_Toc514250104)

[13.3.3 GET\_ORDER\_EVT\_LIST 49](#_Toc514250105)

[13.3.4 GET\_ORDER\_EVT\_LIST\_COUNT 50](#_Toc514250106)

1. PREFACE
   1. Revision history

| Revision | Date | Author | Details |
| --- | --- | --- | --- |
| 0.1 | 09.03.2017 | Gyumyush D. | Initial version |
| 0.2 | 22.03.2017 | Gogolev I. | API for virtual account numbers loading (#CORE-11994) |
| 0.3 | 03.04.2017 | Gogolev I. | API “Customer view information” (#CORE-11723) |
| 0.4 | 06.04.2017 | Kondratyev A. | Credits section |
| 0.5 | 06.04.2017 | Alalykin A. | Visa section |
| 0.6 | 07.04.2017 | Gogolev I. | Added description for all procedures of package - gui\_api\_external\_pkg of the API “Customer view information” (#CORE-12243) |
| 0.7 | 12.04.2017 | Gyumyush D. | Added procedure of card instance delivery status date changing |
| 0.8 | 24.04.2017 | Gogolev I. | Added procedures of getting addresses and contacts information of the customer. Changed output fields of get\_main\_customer\_info cursor |
| 0.9 | 11.05.2017 | Gogolev I. | Added information about gender in to main customer info. Added procedure which returning card features information. |
| 1.0 | 12.05.2017 | Nick S. | Added additional filter by agent\_id |
| 1.1 | 16.05.2017 | Gogolev I. | Added new fields into get\_main\_customer\_info cursor |
| 1.2 | 19.05.2017 | Gogolev I. | Added new fields into get\_main\_cards\_info cursor |
| 1.3 | 15.08.2017 | Man Do | Added new fields into get\_main\_customer\_info cursor.  Added new fields into get\_main\_cards\_info cursor. |
| 1.4 | 21.08.2017 | Man Do | Added new field into get\_main\_cards\_info cursor. |
| 1.5 | 03.09.2017 | Gogolev I. | Added new section – “Transactions” and procedure to it. |
| 1.6 | 07.09.2017 | Gogolev I. | Changed input parameters in procedure get\_transactions\_data |
| 1.7 | 08.09.2017 | Gogolev I. | Changed input parameters in procedure get\_transactions\_data: instead of i\_link\_account\_entity and i\_link\_account\_object\_id was added i\_object\_tab parameter; Corrected meaning of i\_load\_reversals parameter |
| 1.8 | 21.09.2017 | Gogolev I. | Changed sctructure of the input parameter i\_object\_tab in procedure get\_transactions\_data. |
| 1.9 | 25.10.2017 | Gyumyush D. | Added two more procedures add\_bucket and get\_scoring\_data into a new section “Scoring”. |
| 2.0 | 26.10.2017 | Gyumyush D. | Added procedure of GL account numbers and turnovers receiving get\_gl\_account\_numbers\_data |
| 2.1 | 30.10.2017 | Gyumyush D. | Added procedure of GL balancies for account get\_account\_gl\_balancies |
| 2.2 | 17.11.2017 | Gogolev I. | Corrected sections: 5.3, 5.3.3 and added new par. in 5.3 |
| 2.3 | 21.11.2017 | Gogolev I. | Actualization information of sections: 5.3.3 and 5.3.4 |
| 2.4 | 24.11.2017 | Gogolev I. | Actualization information of section: 5.3.3 |
| 2.5 | 29.11.2017 | Gyumyush D. | Added procedure receiving events cursor |
| 2.6 | 04.12.2017 | Gogolev I. | Actualization name of procedure in section 5.3.4 |
| 2.7 | 10.04.2018 | Gogolev I. | Added operations section |
| 2.8 | 16.05.2018 | Fomichev A. | Added payment orders section |
| 2.9 | 16.06.2018 | Pestryakov D. | Methods add\_customer, add\_contract, set\_service\_object, create\_cardholder, issue\_card, reissue\_card, create\_account, add\_account\_object, set\_flexible\_value, add\_person |
| 2.10 | 17.08.2018 | Mikhailov V. | Added overload methods set\_is\_settled of setting flag is\_settled and creation of event EVNT0362. |
| 2.11 | 03.09.2018 | Pestryakov D. | Methods added: get\_object\_flexible\_data, get\_object\_notes, get\_payment\_order, get\_opr\_entries, get\_opr\_clearing\_data, get\_auth\_tag\_values, get\_opr\_participants, get\_opr\_additional\_amount, get\_operations |
| 2.12 | 01.10.2018 | Pestryakov D. | Language is added to get\_object\_notes |
| 2.13 | 24.12.2018 | Nick S. | sttl\_flag\_date is addded |

* 1. Document purpose

SmartVista API Interaction Protocol is a reference manual for developers who are implementing API of the SmartVista solution. This document is written for internal use of BPC and BPC Customers developers. The document describes the content and the structure of the API.

1. SMARTVISTA API INTERACTION OVERVIEW
   1. General concepts

SmartVista API Interaction Protocol (API hereafter) provides a description of the procedures, functions and data formats provided developers an access to the DB infrastructure.

1. CUSTOMERS
   1. Overview

Following methods provide API for work with customer entities and theirs data. Procedures, functions, data structures are described below.

* 1. References

Procedures are placed in packages:

gui\_api\_external\_pkg

* 1. List of PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| gui\_api\_external\_pkg | get\_customer\_list | proc | Return count of record and ref\_cursor |
| gui\_api\_external\_pkg | get\_first\_customer\_id | func | Returning id of the first customer which was find |
| gui\_api\_external\_pkg | get\_main\_customer\_info | proc | Return count of record and ref\_cursor |
| gui\_api\_external\_pkg | get\_customer\_limit\_info | proc | Return count of record and ref\_cursor |
| gui\_api\_external\_pkg | get\_contacts\_customer\_info | proc | Return count of record and ref\_cursor |
| gui\_api\_external\_pkg | get\_addresses\_customer\_info | proc | Return count of record and ref\_cursor |
| prd\_api\_external\_pkg | add\_customer | proc | Add customer |

* + 1. GET\_CUSTOMER\_LIST

Input parameters:

i\_customer\_number – Number of the customer (not mandatory);

i\_customer\_name – Surname and/or first name and/or second of the customer (not mandatory);

i\_customer\_mobile – Mobile phone of the customer (not mandatory);

i\_identity\_type – Type of the identity card of the customer (value of dictionary) (not mandatory);

i\_identity\_series – Serie of the identity card (not mandatory);

i\_identity\_number – Number of the identity card (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| CUSTOMER\_ID | NUMBER(12) | ID of the customer |
| CUSTOMER\_NUMBER | VARCHAR2(200) | Number of the customer |
| CUSTOMER\_NAME | VARCHAR2(4000) | Surname and/or First name and/or Second name of the customer |

This procedure returning main parameters of the customer. Fields which returning are listed higher.

* + 1. GET\_FIRST\_CUSTOMER\_ID

Input parameters:

i\_customer\_number – Number of the customer (not mandatory);

i\_customer\_name – Surname and/or first name and/or second of the customer (not mandatory);

i\_customer\_mobile – Mobile phone of the customer (not mandatory);

i\_identity\_type – Type of the identity card of the customer (value of dictionary) (not mandatory);

i\_identity\_series – Serie of the identity card (not mandatory);

i\_identity\_number – Number of the identity card (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters: Not;

This function returning CUSTOMER\_ID for first customer which was find in customer number ascend sorting order.

* + 1. GET\_MAIN\_CUSTOMER\_INFO

Input parameters:

i\_customer\_number – Number of the customer (not mandatory);

i\_customer\_name – Surname and/or first name and/or second of the customer (not mandatory);

i\_customer\_mobile – Mobile phone of the customer (not mandatory);

i\_identity\_type – Type of the identity card of the customer (value of dictionary) (not mandatory);

i\_identity\_series – Serie of the identity card (not mandatory);

i\_identity\_number – Number of the identity card (not mandatory);

i\_agent\_id – Agent identifier (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| CUSTOMER\_ID | NUMBER(12) | ID of the customer |
| CUSTOMER\_NUMBER | VARCHAR2(200) | Number of the customer |
| CUSTOMER\_NAME | VARCHAR2(4000) | Surname and/or First name and/or Second name of the customer |
| CUSTOMER\_TYPE | VARCHAR2(8) | Type of the customer (value of the dictionary) |
| CUSTOMER\_TYPE\_NAME | VARCHAR2(200) | Name of the customer type |
| COMPANY\_SHORT\_NAME | VARCHAR2(4000) | Short name of company if customer have type ENTTCOMP |
| COMPANY\_FULL\_NAME | VARCHAR2(4000) | Full name of company if customer have type ENTTCOMP |
| DOCUMENT\_TYPE | VARCHAR2(200) | Name of article of the IDTP dictionary |
| ID\_DOCUMENT | VARCHAR2(400) | SERIES and NUMBER of the document of customer |
| PERSON\_AGE | NUMBER(3) | Age of the customer |
| BIRTHDAY | DATE | Birthday of the customer |
| GENDER | VARCHAR2(8) | Gender of the customer (article of the dictionary GNDR) |
| GENDER\_NAME | VARCHAR2(200) | Name of the customer gender |
| HOME\_PHONE\_CONTACT\_ID | NUMBER(12) | Identifier of the contact with communication method “PHONE” |
| HOME\_PHONE\_CONTACT\_TYPE | VARCHAR2(8) | Type of the contact (value of the dictionary) |
| HOME\_PHONE\_COMMUN\_METHOD | VARCHAR2(8) | Communication method (value of the dictionary) |
| HOME\_PHONE | VARCHAR2(200) | Home phone of the customer |
| HOME\_ADDRESS\_ID | NUMBER(12) | Identifier of the home address |
| HOME\_ADDRESS\_TYPE | VARCHAR2(8) | Type of the address (value of the dictionary) |
| HOME\_ADDRESS | VARCHAR2(200) | Home address of the customer |
| BUSINESS\_PHONE\_CONTACT\_ID | NUMBER(12) | Identifier of the contact with communication method “PHONE” |
| BUSINESS\_PHONE\_CONTACT\_TYPE | VARCHAR2(8) | Type of the contact (value of the dictionary) |
| BUSINESS\_PHONE\_COMMUN\_METHOD | VARCHAR2(8) | Communication method (value of the dictionary) |
| BUSINESS\_PHONE | VARCHAR2(200) | Business phone of the customer |
| BUSINESS\_ADDRESS\_ID | NUMBER(12) | Identifier of the business address |
| BUSINESS\_ADDRESS\_TYPE | VARCHAR2(8) | Type of the address (value of the dictionary) |
| BUSINESS\_ADDRESS | VARCHAR2(200) | Business address of the customer |
| MOB\_PHONE\_CONTACT\_ID | NUMBER(12) | Identifier of the contact with communication method “MOBILE PHONE” |
| MOB\_PHONE\_CONTACT\_TYPE | VARCHAR2(8) | Type of the contact (value of the dictionary) |
| MOB\_PHONE\_COMMUN\_METHOD | VARCHAR2(8) | Communication method (value of the dictionary) |
| MOBILE\_PHONE | VARCHAR2(200) | Mobile phone of the customer |
| FAX\_CONTACT\_ID | NUMBER(12) | Identifier of the contact with communication method “FAX” |
| FAX\_CONTACT\_TYPE | VARCHAR2(8) | Type of the contact (value of the dictionary) |
| FAX\_COMMUN\_METHOD | VARCHAR2(8) | Communication method (value of the dictionary) |
| FAX | VARCHAR2(200) | Fax of the customer |
| EMAIL\_CONTACT\_ID | NUMBER(12) | Identifier of the contact with communication method “EMAIL” |
| EMAIL\_CONTACT\_TYPE | VARCHAR2(8) | Type of the contact (value of the dictionary) |
| EMAIL\_COMMUN\_METHOD | VARCHAR2(8) | Communication method (value of the dictionary) |
| EMAIL | VARCHAR2(200) | Email of the customer |
| START\_DATE | DATE | First date of the registration of customer |
| TERMINATION\_DATE | DATE | Termination date of the customer |
| INST\_ID | NUMBER(4) | ID of the current term institution of customer |
| AGENT\_ID | NUMBER(8) | ID of the current term agent of customer |
| NATIONALITY\_CODE | VARCHAR2(3) | Customer nationality country code |
| NATIONALITY\_NAME | VARCHAR2(200) | Customer nationality country name |
| FLEXIBLE\_FIELDS | - | If entity CUSTOMER link with flexible fields on current system then they will return |

This procedure returning main parameters of the customer. Fields which returning are listed higher.

* + 1. GET\_CUSTOMER\_LIMIT\_INFO

Input parameters:

i\_customer\_id – Identifier of the customer (mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| ID | NUMBER(8) | Identifier of the attribute |
| ATTR\_SYSTEM\_NAME | VARCHAR2(200) | Name of the attribute |
| CUSTOMER\_LIMIT\_NAME | VARCHAR2(200) | Label of the attribute |
| LIMIT\_CURRENCY | VARCHAR2(3) | Code of currency |
| LIMIT\_SUM | NUMBER(22, 4) | Value of the limit |
| SUM\_REMINDER | NUMBER(22, 4) | Rest of the available funds |

This procedure returning main parameters of the customer level limits (attributes for CUSTOMER MAINTENANCE SERVICE TYPE). Fields which returning are listed higher.

* + 1. GET\_CONTACTS\_CUSTOMER\_INFO

Input parameters:

i\_customer\_id – Identifier of the customer (not mandatory\*);

i\_contact\_id - Identifier of the cotact (not mandatory\*);

\*Remark: At least one parameter must be set.

i\_contact\_type – Type of the contact, value from dictionary (not mandatory);

i\_commun\_method – Method of the communication, value from dictionary (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| CUSTOMER\_ID | NUMBER(12) | Identifier of the customer |
| CONTACT\_ID | NUMBER(12) | Identifier of the contact |
| CONTACT\_TYPE | VARCHAR2(8) | Type of the contact (value of the dictionary) |
| CONTACT\_TYPE\_NAME | VARCHAR2(200) | Name of the contact type |
| COMMUNICATION\_METHOD | VARCHAR2(8) | Method of the communication (value of the dictionary) |
| COMMUNICATION\_METHOD\_NAME | VARCHAR2(200) | Name of the communication method |
| COMMUNICATION\_STRING | VARCHAR2(200) | String of the communication data |
| START\_DATE | DATE | Start date of communication data validity |
| END\_DATE | DATE | End date of communication data validity |
| FLEXIBLE\_FIELDS | - | If entity CONTACT link with flexible fields on current system then they will return |

This procedure returning contacts information of the customer. Fields which returning are listed higher.

* + 1. GET\_ADDRESSES\_CUSTOMER\_INFO

Input parameters:

i\_customer\_id – Identifier of the customer (not mandatory\*);

i\_address\_id – Identifier of the address (not mandatory\*);

\*Remark: At least one parameter must be set.

i\_address\_type - Type of the address, value from dictionary (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| CUSTOMER\_ID | NUMBER(12) | Identifier of the customer |
| ADDRESS\_ID | NUMBER(12) | Identifier of the address |
| ADDRESS\_TYPE | VARCHAR2(8) | Type of the address (value of the dictionary) |
| ADDRESS\_TYPE\_NAME | VARCHAR2(200) | Name of the address type |
| ADDRESS\_STRING | VARCHAR2(200) | Full addressed string |
| COUNTRY\_CODE | VARCHAR2(3) | Numerical code of the country |
| COUNTRY\_CODE\_NAME | VARCHAR2(3) | Char code of the country |
| COUNTRY\_NAME | VARCHAR2(200) | Full country name |
| REGION | VARCHAR2(200) | Region name |
| CITY | VARCHAR2(200) | City |
| STREET | VARCHAR2(200) | Street |
| HOUSE | VARCHAR2(200) | House |
| APARTMENT | VARCHAR2(200) | Office/apartment |
| POSTAL\_CODE | VARCHAR2(10) | Postal code |
| REGION\_CODE | VARCHAR2(20) | Region code |
| LATITUDE | NUMBER(10, 7) | Geographic coordinate - Latitude (N) |
| LONGITUDE | NUMBER(10, 7) | Geographic coordinate - Longitude (W) |
| INST\_ID | NUMBER(4) | Owner institution identifier |
| INST\_NAME | VARCHAR2(200) | Name of the institution |
| PLACE\_CODE | VARCHAR2(200) | Place code |
| FLEXIBLE\_FIELDS | - | If entity ADDRESS link with flexible fields on current system then they will return |

This procedure returning addresses information of the customer. Fields which returning are listed higher.

* + 1. ADD\_CUSTOMER

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| o\_id | out | number(12) | Customer Id |
| o\_seqnum | out | number(4) | Customer seq.number |
| i\_entity\_type | in | varchar2(8) | Entity type |
| i\_object\_id | in | number(16) | Object id |
| io\_customer\_number | in out | varchar2(200) | Customer number |
| i\_inst\_id | in | number(4) | Institution id |
| i\_category | in | varchar2(8) | Category |
| i\_relation | in | varchar2(8) | Relation |
| i\_resident | in | number(1) | Resident |
| i\_nationality | in | varchar2(3) | Nationality |
| i\_credit\_rating | in | varchar2(8) | Credit rating |
| i\_money\_laundry\_risk | in | varchar2(8) | Money laundry risk |
| i\_money\_laundry\_reason | in | varchar2(8) | Money laundry reason |
| i\_status | in | varchar2(8) | Customer status |
| i\_ext\_entity\_type | in | varchar2(8) | External entity type |
| i\_ext\_object\_id | in | number(16) | External object id |
| i\_product\_type | in | varchar2(8) | Product type |
| i\_employment\_status | in | varchar2(8) | Employment status |
| i\_employment\_period | in | varchar2(8) | Employment period |
| i\_residence\_type | in | varchar2(8) | Residence type |
| i\_marital\_status | in | varchar2(8) | Marital status |
| i\_marital\_status\_date | in | date | Marital status date |
| i\_income\_range | in | varchar2(8) | Income range |
| i\_number\_of\_children | in | varchar2(8) | Number of children |

Creates new customer and returns its identifier and sequential number.

1. ACCOUNTS
   1. OVERVIEW

Following methods provide API for work with account entities and theirs data. Procedures, functions, data structures are described below.

* 1. REFERENCES

Procedures are placed in packages:

gui\_api\_external\_pkg

* 1. LIST OF PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| gui\_api\_external\_pkg | get\_main\_billing\_info | proc | Return count of record and ref\_cursor |
| acc\_api\_external\_pkg | create\_account | proc | Creates account |
| acc\_api\_external\_pkg | add\_account\_object | proc | Attaches object to account |

* + 1. GET\_MAIN\_BILLING\_INFO

Input parameters:

i\_customer\_id – Identifier of the customer (mandatory);

i\_agent\_id – Agent identifier (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| SEQ | NUMBER | Number of the line |
| ACCOUNT\_ID | NUMBER(12) | Identifier of the account |
| ACCOUNT\_TYPE | VARCHAR2(8) | Account type (article of the dictionary ACTP) |
| ACCOUNT\_TYPE\_NAME | VARCHAR2(200) | Name of article of the ACTP dictionary |
| ACCOUNT\_STATUS | VARCHAR2(8) | Account status (article of the dictionary ACST) |
| ACCOUNT\_STATUS\_NAME | VARCHAR2(200) | Name of article of the ACST dictionary |
| DUE\_DATE | DATE | Due date (for credit) |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Number of account |
| AGENT\_ID | NUMBER(8) | ID of the term account agent |
| AGENT\_NAME | VARCHAR2(200) | Agent name |
| CUSTOMER\_ADDRESS | VARCHAR2(200) | Address of the customer |
| STATEMENT\_DELIVERY | VARCHAR2(200) | Statement delivery |
| RETURNED\_STATUS | VARCHAR2(200) | Returned status |
| START\_DATE | DATE | Date of the account registration |
| TERM\_DATE | DATE | Termination date of the account |
| FLEXIBLE\_FIELDS | - | If entity ACCOUNT link with flexible fields on current system then they will return |

This procedure returning main parameters of the customer accounts. Fields which returning are listed higher.

* + 1. CREATE\_ACCOUNT

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| o\_id | out | number(12) | id |
| io\_split\_hash | in out | number(4) | split hash |
| i\_account\_type | in | varchar2(8) | account type |
| io\_account\_number | in out | varchar2(32) | account number |
| i\_currency | in | varchar2(3) | currency |
| i\_inst\_id | in | number(4) | institution id |
| i\_agent\_id | in | number(8) | agent id |
| i\_status | in | varchar2(8) | status |
| i\_contract\_id | in | number(12) | contract id |
| i\_customer\_id | in | number(12) | customer id |
| i\_customer\_number | in | varchar2(200) | customer number |

Creates new account

* + 1. ADD\_ACCOUNT\_OBJECT

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| i\_account\_id | in | number(12) | account id |
| i\_entity\_type | in | varchar2(8) | entity type |
| i\_object\_id | in | number(16) | object id |
| i\_usage\_order | in | number(4) | usage order |
| i\_is\_pos\_default | in | number(1) | is pos default |
| i\_is\_atm\_default | in | number(1) | is atm default |
| i\_is\_atm\_currency | in | number(1) | is atm currency |
| i\_is\_pos\_currency | in | number(1) | is pos currency |
| i\_account\_seq\_number | in | number(2) | account seq number |
| o\_account\_object\_id | out | number(16) | account object id |

Attach object to the account

1. TRANSACTIONS
   1. OVERVIEW

Following methods provide API for work with transactions objects and theirs data. Procedures, functions, data structures are described below.

* 1. REFERENCES

Procedures are placed in packages:

acc\_api\_external\_pkg

* 1. LIST OF PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| acc\_api\_external\_pkg | get\_transactions\_data | proc | Return count of record and ref\_cursor |
| acc\_api\_external\_pkg | get\_gl\_account\_numbers\_data | proc | Returns count of records and ref\_curssor with GL account data |
| acc\_api\_external\_pkg | get\_active\_accounts\_for\_period | proc | Return count of records and information of account properties for active accounts |
| acc\_api\_external\_pkg | get\_link\_account\_balances | proc | Return count of records and data of the link accounts balances for main account |

* + 1. GET\_TRANSACTIONS\_DATA

Input parameters:

i\_inst\_id – Identifier of the institution (mandatory);

i\_date\_type - The type of the date (dictionary article) for the selection of transactions (mandatory);

i\_start\_date - Date begin (for default – current date on day begin) (not mandatory);

i\_end\_date - Date end (for default – current date on day end) (not mandatory);

i\_balance\_type - Type of the balance account (dictionary article) (not mandatory);

i\_account\_number - Number of the account (not mandatory);

i\_fees - Load fees transactions (Yes/No) Yes – only fees; No – exclude fees from select; Null – not check (not mandatory);

i\_gl\_accounts - Load transactions with GL-accounts (Yes/No) Yes – only GL; No – exclude GL from select; Null – not check (not mandatory);

i\_load\_reversals - Load a reversals (Yes/No): Yes (or null) – include reversals; No – exclude reversals from select;

i\_object\_tab – Table of objects which use in select of transactions, with defined structure: level\_type (entity type of level for define of link between current object and transactions), entity\_type (entity type of object), object\_id (table of the objects identifier) (not mandatory);

i\_array\_balance\_type\_id - Identifier of account balances array (not mandatory);

i\_array\_trans\_type\_id - Identifier of transaction types array (not mandatory);

i\_array\_settl\_type\_id - Identifier of settlement types array (not mandatory);

i\_array\_operations\_type\_id - Identifier of operation types array (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| oper\_id | number(16) | opr\_operation.id |
| oper\_type | varchar2(8) | opr\_operation.oper\_type |
| oper\_reason | varchar2(8) | opr\_operation.oper\_reason |
| is\_reversal | number(1) | opr\_operation.is\_reversal |
| msg\_type | varchar2(8) | opr\_operation.msg\_type |
| sttl\_type | varchar2(8) | opr\_operation.sttl\_type |
| oper\_amount | number(22,4) | opr\_operation.oper\_amount |
| oper\_currency | varchar2(3) | opr\_operation.oper\_currency |
| oper\_date | date | opr\_operation.oper\_date |
| host\_date | date | opr\_operation.host\_date |
| unhold\_date | date | opr\_operation.unhold\_date |
| oper\_sttl\_date | date | opr\_operation.sttl\_date |
| macros\_id | number(16) | acc\_macros.id |
| macros\_account\_id | number(12) | acc\_macros.account\_id |
| macros\_amount | number(22,4) | acc\_macros.amount |
| macros\_currency | varchar2(3) | acc\_macros.currency |
| macros\_amount\_purpose | varchar2(8) | acc\_macros.amount\_purpose |
| macros\_posting\_date | date | acc\_macros.posting\_date |
| macros\_conversion\_rate | number | acc\_macros.conversion\_rate |
| macros\_conversion\_rate\_id | number(8) | acc\_macros.conversion\_rate\_id (new field from p.1.1.1) |
| transaction\_id | number(16) | acc\_entry.transaction\_id |
| transaction\_type | varchar2(8) | acc\_entry.transaction\_type |
| debt\_entry\_id | number(16) | acc\_entry.id (for debt) |
| debt\_account\_id | number(12) | acc\_entry.account\_id (for debt) |
| debt\_account\_number | varchar2(32) | acc\_account.account\_number (for debt) |
| debt\_account\_currency | varchar2(3) | acc\_account.currency(for debt) |
| debt\_balance\_currency | varchar2(3) | acc\_balance.currency(for debt) |
| debt\_amount | number(22,4) | acc\_entry.amount (for debt) |
| debt\_currency | varchar2(3) | acc\_entry.currency (for debt) |
| debt\_balance\_type | varchar2(8) | acc\_entry.balance\_type (for debt) |
| debt\_balance\_impact | number(1) | acc\_entry.balance\_impact (for debt) |
| debt\_balance | number(22,4) | acc\_entry.balance (for debt) |
| debt\_posting\_date | date | acc\_entry.posting\_date (for debt) |
| debt\_sttl\_date | date | acc\_entry.sttl\_date (for debt) |
| credit\_entry\_id | number(16) | acc\_entry.id (for credit) |
| credit\_account\_id | number(12) | acc\_entry.account\_id (for credit) |
| credit\_account\_number | varchar2(32) | acc\_account.account\_number (for credit) |
| credit\_account\_currency | varchar2(3) | acc\_account.currency(for credit) |
| credit\_balance\_currency | varchar2(3) | acc\_balance.currency(for credit) |
| credit\_amount | number(22,4) | acc\_entry.amount (for credit) |
| credit\_currency | varchar2(3) | acc\_entry.currency (for credit) |
| credit\_balance\_type | varchar2(8) | acc\_entry.balance\_type (for credit) |
| credit\_balance\_impact | number(1) | acc\_entry.balance\_impact (for credit) |
| credit\_balance | number(22,4) | acc\_entry.balance (for credit) |
| credit\_posting\_date | date | acc\_entry.posting\_date (for credit) |
| credit\_sttl\_date | date | acc\_entry.sttl\_date (for credit) |

This procedure returning main parameters of the transactions linked with entries, operations and accounts entities. Fields which returning are listed higher.

* + 1. GET\_GL\_ACCOUNT\_NUMBERS\_DATA

Input parameters:

i\_inst\_id – Identifier of the institution (mandatory);

i\_start\_date - Date begin (for default – current date on day begin) (not mandatory);

i\_end\_date - Date end (for default – current date on day end) (not mandatory).

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| account\_number | varchar2 (32) | Account number |
| card\_mask | varchar2 (24) | Card masked number |
| customer\_number | varchar2 (200) | Customer number |
| national\_id | varchar2 (200) | National ID |
| gl\_account\_number | varchar2 (32) | GL account number |
| macros\_type\_id | Number (4) |  |
| oper\_type | varchar2 (8) |  |
| oper\_date | Date | Transaction Date |
| posting\_date | Date | Posting Date |
| amount | NUMBER (22,4) |  |
| due\_date | Date |  |
| aging | number(2) | Aging period |

Procedure gathers data about GL accounts, entries and operations related with their.

* + 1. GET\_ACTIVE\_ACCOUNTS\_FOR\_PERIOD

Input parameters:

i\_inst\_id – Identifier of the institution (mandatory);

i\_date\_type - The type of the date (dictionary article) for the selection data in acc\_entry (mandatory);

i\_start\_date - Date begin (for default – current date on day begin, not set in this procedure) (mandatory);

i\_end\_date - Date end (for default – current date on day end, not set in this procedure) (mandatory);

i\_account\_id – Account identifier (optional);

i\_gl\_accounts - Unload GL-accounts (Yes/No) Yes – only GL; No – exclude GL from select; Null – not check (optional);

i\_mask\_error - Mask error (Yes/No) (for default – No) (optional).

Output parameters:

o\_row\_count – count of records;

o\_ref\_cursor – Result cursor with data of accounts.

|  |  |  |
| --- | --- | --- |
| Accounts collection (o\_ref\_cursor) | | |
| Name | Type | Description |
| account\_id | number(12) | Account identifier |
| account\_number | varchar2 (32) | Account number |
| split\_hash | number(4) | Hash value to split further processing |
| card\_mask | varchar2 (24) | Card masked number |
| customer\_number | varchar2 (200) | Customer number |
| national\_id\_type | varchar2 (8) | Type of customer person identity card |
| national\_id | varchar2 (200) | National ID |
| product\_id | number(8) | ID of the product |
| product\_number | varchar2 (200) | The code of the product in SV |
| loan\_for\_year | number(4) | Number of loans per year |
| aging | number(2) | Aging period |

* + 1. GET\_LINK\_ACCOUNT\_BALANCES

Input parameters:

i\_date\_type - The type of the date (dictionary article) for the selection data in acc\_entry (mandatory);

i\_start\_date - Date begin (for default – current date on day begin, not set in this procedure) (mandatory);

i\_end\_date - Date end (for default – current date on day end, not set in this procedure) (mandatory);

i\_account\_id – Account identifier (mandatory);

i\_gl\_accounts - Unload balance info of the GL-accounts (Yes/No) Yes – only GL; No – exclude GL from select; Null – not check (optional);

i\_array\_link\_account\_numbers – Array of the link accounts numbers which balances will be calculate (optional);

i\_mask\_error - Mask error (Yes/No) (for default – No) (optional).

Output parameters:

o\_row\_count – count of records;

o\_ref\_cursor - Result cursor with link account balancies.

|  |  |  |
| --- | --- | --- |
| Account balances cursor (o\_ref\_cursor) | | |
| Name | Type | Description |
| account\_id | number(12) | Account identifier |
| link\_account\_id | number(12) | Link account identifier |
| link\_account\_number | varchar2(32) | Link account number |
| balance\_amount | number(22,4) | Link account balance for account\_id |

* + 1. SET\_IS\_SETTLED

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| i\_entry\_id | in | number(16) | entry id |
| i\_is\_settled | in | number(1) | flag CBS settled entry |
| i\_inst\_id | in | number(4) | institution id |
| i\_split\_hash | in | number(4) | split hash |
| i\_sttl\_flag\_date | in | date | Settled flag date |

Sets acc\_entry.is\_settled flag, creates event EVNT0362 for one entry

* + 1. SET\_IS\_SETTLED

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| i\_entry\_id\_tab | in | collection of number(16) | list of entry id |
| i\_is\_settled | in | number(1) | flag CBS settled entry |
| i\_inst\_id | in | collection of number(4) | list of institution id |
| i\_split\_hash | in | collection of number(4) | list of split hash |
| i\_sttl\_flag\_date | in | date | Settled flag date |

Sets acc\_entry.is\_settled flag, creates event EVNT0362 for list of entries

* + 1. SET\_IS\_SETTLED

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| i\_operation\_id\_tab | in | collection of number | list of operation id |
| i\_is\_settled | in | number(1) | flag CBS settled entry |
| i\_inst\_id | in | collection of number(4) | list of institution id |
| i\_split\_hash | in | collection of number(4) | list of split hash |
| i\_sttl\_flag\_date | in | date | Settled flag date |

Sets acc\_entry.is\_settled flag, creates event EVNT0362 for list of operations

* + 1. GET\_OPR\_ENTRIES

Input parameters:

i\_oper\_id – Identifier of operation (mandatory)

i\_array\_balance\_type\_id - Identifier of account balances array

i\_array\_trans\_type\_id - Identifier of transaction types array

i\_array\_settl\_type\_id - Identifier of settlement types array

i\_array\_operations\_type\_id - Identifier of operation types array

Output parameters:

o\_ref\_cursor - Result cursor

Resultset of the cursor matches sections “transaction”+”entry” of clearing file structure in svxp\_cbs\_eng.docx

To fetch records of this cursor type acc\_api\_external\_pkg.t\_entries\_tab should be used.

1. CARD
   1. OVERVIEW

Following methods provide API for work with card entities and theirs data. Procedures, functions, data structures are described below.

* 1. REFERENCES

Procedures are placed in packages:

gui\_api\_external\_pkg

iss\_api\_card\_inst\_extended\_pkg

* 1. LIST OF PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| gui\_api\_external\_pkg | get\_main\_cards\_info | proc | Return count of record and ref\_cursor |
| gui\_api\_external\_pkg | get\_customer\_marketing\_info | proc | Return count of record and ref\_cursor |
| iss\_api\_card\_inst\_extended\_pkg | change\_delivery\_status\_date | proc | Procedure updates date of delivery status changing |
| gui\_api\_external\_pkg | get\_card\_type\_feature | proc | Return count of record and ref\_cursor |
| iss\_api\_external\_pkg | create\_cardholder | proc | Creates cardholder |
| iss\_api\_external\_pkg | issue\_card | proc | Creates new card |
| iss\_api\_external\_pkg | reissue\_card | proc | Reissue of card |

* + 1. GET\_MAIN\_CARDS\_INFO

Input parameters:

i\_customer\_id – Identifier of the customer (mandatory);

i\_agent\_id – Agent identifier (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| SEQ | NUMBER | Number of the line |
| CARD\_ACCOUNT\_TYPE | VARCHAR2(8) | Type of the card account (article of the dictionary ACTP) |
| CARD\_NETWORK | VARCHAR2(200) | Network of the card |
| CARD\_TYPE\_NAME | VARCHAR2(200) | Name of card type |
| CARD\_MASK | VARCHAR2(24) | Mask of the card |
| CARD\_NUMBER | VARCHAR2(24) | Number of the card |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Number of the card account |
| PRODUCT\_ID | NUMBER(8) | Identificator of the product |
| CONTRACT\_TYPE | VARCHAR2(8) | Contract type. Describe business area where product could be used. |
| CONTRACT\_TYPE\_NAME | VARCHAR2(200) | Name of the field Contract Type |
| PRODUCT\_TYPE | VARCHAR2(8) | Type of the card product type (article of the dictionary PRDT) |
| PRODUCT\_TYPE\_NAME | VARCHAR2(200) | Name of article of the PRDT dictionary |
| PRODUCT\_NUMBER | VARCHAR2(200) | External product number |
| PRODUCT\_LABEL | VARCHAR2(200) | Product label |
| CARD\_STATE\_NAME | VARCHAR2(200) | Name of the article CSTE dictionary |
| CARD\_CATEGORY\_NAME | VARCHAR2(200) | Name of the article CRCG dictionary |
| CARDHOLDER\_NAME\_INSTANCE | VARCHAR2(200) | The cardholder name of card instance |
| ISS\_DATE | DATE | Date of the issue card |
| REISSUE\_REASON | VARCHAR2(200) | Name of the article EVNT dictionary |
| EXPIR\_DATE | DATE | Card expiration date |
| CARD\_STATUS\_CODE | VARCHAR2(8) | Card status (article of the dictionary CSTS) |
| CARD\_STATUS\_DESCR | VARCHAR2(200) | Name of the article CSTS dictionary |
| EXTEND\_IDENTIFIED\_TYPE | VARCHAR2(200) | Name of article of the IDTP dictionary |
| EXTEND\_IDENTIFIED\_NUM | VARCHAR2(400) | SERIES and NUMBER of the document |
| BLANK\_TYPE | VARCHAR2(200) | Name of blank type |
| CARDHOLDER\_NAME | VARCHAR2(200) | Name of the cardholder |
| BIRTHDAY | DATE | Customer’s birthday |
| ID\_TYPE | VARCHAR2(200) | Customer’s ID type |
| ID\_NUMBER | VARCHAR2(200) | Customer’s ID number |
| DELIVERY\_CHANNEL | VARCHAR2(200) | Card delivery channel  (article of the dictionary CRDC) |
| DELIVERY\_STATUS | VARCHAR2(200) | Card delivery status (article of the dictionary CRDS) |
| STATEMENT\_DELIVERY | VARCHAR2(200) | Statement delivery method  (article of the dictionary STDM) |
| FLEXIBLE\_FIELDS | - | If entity CARD link with flexible fields on current system then they will return |

This procedure returning main parameters of the customer cards. Fields which returning are listed higher.

* + 1. GET\_CUSTOMER\_MARKETING\_INFO

Input parameters:

i\_customer\_id – Identifier of the customer (mandatory);

i\_months\_ago – Counts of months ago with current day (mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| CARD\_TYPE\_ID | NUMBER(4) | Card type identifier |
| CARD\_TYPE\_NAME | VARCHAR2(200) | Name of card type |
| CARD\_NUMBER | VARCHAR2(24) | Number of the card |
| CARD\_ACCOUNT\_NUMBER | VARCHAR2(32) | Number of the card account |
| CARD\_ACCOUNT\_TYPE | VARCHAR2(8) | Account type of the card (article of the dictionary ACTP) |
| CARD\_ACCOUNT\_TYPE\_NAME | VARCHAR2(200) | Name of article of the ACTP dictionary |
| IS\_CREDIT\_FLAG | NUMBER(1) | A sign that the card account is a credit account |
| OPER\_TYPE | VARCHAR2(8) | Type of operation (article of the dictionary OPTP) |
| OPER\_TYPE\_NAME | VARCHAR2(200) | Name of article of the OPTP dictionary |
| CURRENCY\_NAME | VARCHAR2(3) | ISO alpha operation currency code |
| OPER\_AMOUNT | NUMBER(22, 4) | Operation amount in operation currency |

This procedure returning main information about operation which were committed using customer cards. Fields which returning are listed higher.

* + 1. CHANGE\_DELIVERY\_STATUS\_DATE

Input parameters:

i\_entity\_type – entity type for what delivery status has been changed, ENTT dictionary (varchar2(8));

i\_object\_id – unique identifier of entity type object (number(16));

i\_event\_type – event type of delivery status changing, EVNT dictionary (varchar2(8));

i\_initiator – initiator, ENSI dictionary (varchar2(8));

i\_status – value of status what should be change date, CRDS dictionary (varchar2(8));

i\_event\_date – date of delivery status event (date).

Output parameters:

No parameters.

Actions explanation:

Procedure changes event date of changing delivery status of card instance. Updated record should have entity type, object identifier, event type, initiator and card delivery status like passed into procedure.

* + 1. GET\_CARD\_TYPE\_FEATURE

Input parameters:

i\_card\_number – Number of the card (mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| CARD\_FEATURE | VARCHAR2(8) | Type of the card features (article of the dictionary CFCH) |
| FEATURE\_DESCRIPTION | VARCHAR2(2000) | Description of the card features |

This procedure returning features of the customer card. Fields which returning are listed higher.

* + 1. CREATE\_CARDHOLDER

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| o\_id | out | number(12) | Cardholder identifier |
| i\_customer\_id | in | number(12) | Customer id |
| i\_person\_id | in | number(12) | Person id |
| i\_cardholder\_name | in | varchar2(200) | Cardholder name |
| i\_relation | in | varchar2(8) | Relation |
| i\_resident | in | number(1) | Resident |
| i\_nationality | in | varchar2(3) | Nationality |
| i\_marital\_status | in | varchar2(8) | Marital status |
| io\_cardholder\_number | in out | varchar2(200) | Cardholder number |
| i\_inst\_id | in | number(4) | Institution id |

Creates cardholder

* + 1. ISSUE\_CARD

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| o\_id | out | number(12) | Card id |
| io\_card\_number | in out | varchar2(24) | Card number |
| o\_card\_instance\_id | out | number(12) | Card instance id |
| i\_inst\_id | in | number(4) | Institution id |
| i\_agent\_id | in | number(8) | Agent id |
| i\_contract\_id | in | number(12) | Contract id |
| i\_cardholder\_id | in | number(16) | Cardholder id |
| i\_card\_type\_id | in | number(4) | Card type id |
| i\_customer\_id | in | number(16) | Customer id |
| i\_category | in | varchar2(8) | Category |
| i\_start\_date | in | date | Start date |
| io\_expir\_date | in out | date | Expiration date |
| i\_cardholder\_name | in | varchar2(200) | Cardholder name |
| i\_company\_name | in | varchar2(200) | Company name |
| i\_perso\_priority | in | varchar2(8) | Personalization priority |
| i\_service\_id | in | number(8) | Service id |
| i\_icc\_instance\_id | in | number(12) | Icc instance id |
| i\_delivery\_channel | in | varchar2(8) | Delivery channel |
| i\_blank\_type\_id | in | number(4) | Blank type id |
| i\_seq\_number | in | number(4) | Seq number |
| i\_status | in | varchar2(8) | Status |
| i\_state | in | varchar2(8) | State |
| i\_iss\_date | in | date | Iss date |
| i\_preceding\_instance\_id | in | varchar2(8) | Preceding instance id |
| i\_reissue\_reason | in | varchar2(8) | Reissue reason |
| i\_reissue\_date | in | date | Reissue date |
| i\_pin\_request | in | varchar2(8) | Pin request |
| i\_embossing\_request | in | varchar2(8) | Embossing request |
| i\_delivery\_status | in | varchar2(8) | Delivery status |
| i\_embossed\_surname | in | varchar2(200) | Embossed surname |
| i\_embossed\_first\_name | in | varchar2(200) | Embossed first name |
| i\_embossed\_second\_name | in | varchar2(200) | Embossed second name |
| i\_embossed\_title | in | varchar2(8) | Embossed title |
| i\_embossed\_line\_additional | in | varchar2(200) | Embossed line additional |
| i\_supplementary\_info\_1 | in | varchar2(200) | supplementary info 1 |
| i\_cardholder\_photo\_file\_name | in | varchar2(255) | Cardholder photo file name |
| i\_cardholder\_sign\_file\_name | in | varchar2(255) | Cardholder sign file name |
| i\_pin\_mailer\_request | in | varchar2(8) | Pin mailer request |

* + 1. REISSUE\_CARD

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| i\_card\_number | in | varchar2(24) | card number |
| io\_seq\_number | in out | number(4) | seq number |
| io\_card\_number | in out | varchar2(24) | card number |
| i\_command | in | varchar2(8) | command |
| i\_agent\_id | in | number(8) | agent id |
| i\_contract\_id | in | number(12) | contract id |
| i\_card\_type\_id | in | number(4) | card type id |
| i\_category | in | varchar2(8) | category |
| i\_start\_date | in | date | start date |
| i\_start\_date\_rule | in | varchar2(8) | start date rule |
| io\_expir\_date | in out | date | expir date |
| i\_expir\_date\_rule | in | varchar2(8) | expir date rule |
| i\_cardholder\_name | in | varchar2(200) | cardholder name |
| i\_company\_name | in | varchar2(200) | company name |
| i\_perso\_priority | in | varchar2(8) | perso priority |
| i\_pin\_request | in | varchar2(8) | pin request |
| i\_pin\_mailer\_request | in | varchar2(8) | pin mailer request |
| i\_embossing\_request | in | varchar2(8) | embossing request |
| i\_delivery\_channel | in | varchar2(8) | delivery channel |
| i\_blank\_type\_id | in | number(4) | blank type id |
| i\_reissue\_reason | in | varchar2(8) | reissue reason |
| i\_reissue\_date | in | date | reissue date |
| i\_clone\_optional\_services | in | number(1) | clone optional services |
| i\_delivery\_status | in | varchar2(8) | delivery status |
| i\_embossed\_surname | in | varchar2(200) | embossed surname |
| i\_embossed\_first\_name | in | varchar2(200) | embossed first name |
| i\_embossed\_second\_name | in | varchar2(200) | embossed second name |
| i\_embossed\_title | in | varchar2(8) | embossed title |
| i\_embossed\_line\_additional | in | varchar2(200) | embossed line additional |
| i\_supplementary\_info\_1 | in | varchar2(200) | supplementary info |
| i\_cardholder\_photo\_file\_name | in | varchar2(255) | cardholder photo file name |
| i\_cardholder\_sign\_file\_name | in | varchar2(255) | cardholder sign file name |
| i\_card\_uid | in | varchar2(200) | card uid |

1. NAMING FORMAT
   1. OVERVIEW

Following methods provide API for work with naming format and theirs data. Procedures, functions, data structures are described below.

* 1. REFERENCES

Procedures are placed in packages:

itf\_api\_naming\_pkg

* 1. LIST OF PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| itf\_api\_naming\_pkg | import\_pool\_value | proc | Insert one value into table rul\_name\_index\_pool |
| itf\_api\_naming\_pkg | import\_pool\_values | proc | Insert collection of the values into table rul\_name\_index\_pool |

* + 1. IMPORT\_POOL\_VALUE

Input parameters:

i\_index\_range\_id - Index range identifier;

i\_value - Index value (value which use in naming format Operational rules => Naming);

Output parameters: Not;

This procedure to add one value into table rul\_name\_index\_pool. If this value is present in table duplication does not occur for pointed index and exception don’t generated for this event.

* + 1. IMPORT\_POOL\_VALUES

Input parameters:

i\_index\_range\_id - Index range identifier;

i\_values – collection of the index values (see 5.3.1);

Output parameters: Not;

This procedure to add collection values into table rul\_name\_index\_pool. Duplication does not occur for pointed index and exception don’t generated for this event.

1. CREDITS
   1. Overview

Following methods provide API for credit statement. Procedures, functions, data structures are described below.

* 1. References

Procedures are placed in packages:

crd\_api\_external\_pkg

gui\_api\_external\_pkg

* 1. List of PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| crd\_api\_external\_pkg | account\_statement | proc | Return ref\_cursor |
| crd\_api\_external\_pkg | loyalty\_points\_sum | proc | Return ref\_cursor |
| crd\_api\_external\_pkg | credit\_service\_limits | proc | Return ref\_cursor |
| crd\_api\_external\_pkg | account\_card | proc | Return ref\_cursor |
| crd\_api\_external\_pkg | due\_amounts | proc | Return ref\_cursor |
| crd\_api\_external\_pkg | transactions | proc | Return ref\_cursor |
| gui\_api\_external\_pkg | get\_customer\_crd\_invoice\_info | proc | Return count of record and ref\_cursor |
| gui\_api\_external\_pkg | get\_crd\_invoice\_payments\_info | proc | Return count of record and ref\_cursor |

* + 1. ACCOUNT\_STATEMENT

Input parameters:

i\_inst\_id – Institute identifier;

i\_account\_number – Number of account (not mandatory);

i\_invoice\_id – Invoice identifier (not mandatory);

One of i\_account\_number or i\_invoice\_id should be filled.

Output parameters:

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Number of account |
| BILLING\_DATE | DATE | Statement generation date |
| START\_PERIOD | DATE | Start date of invoice period |
| END\_PERIOD | DATE | End date of invoice period |
| SETTLEMENT\_DATE | VARCHAR2(2) | Payment Due Date (Day) |
| WITHDRAW\_DATE | DATE | Payment Due Date |
| PAYMENT\_AMOUNT | NUMBER(22,4) | Total Amount Due |
| OVERDUE\_PRINCIPAL\_AMOUNT | NUMBER(22,4) | Overdue principal debt amount for delinquent account. |
| OVERDUE\_INTEREST | NUMBER(22,4) | Overdue Interest debt amount for delinquent account. |
| CARRY\_OVER\_LAST\_MONTH | NUMBER(22,4) | Overdue debt amount plus Interest for non-delinquent account |
| BILLING\_AMT | NUMBER(22,4) | Total transaction for which authorization is done in billing cycle and settlement received by statement generation date |
| MIN\_AMOUNT\_DUE | NUMBER(22,4) | Mandatory Amount Due |
| ADVANCE\_PAYMENT\_AMOUNT | NUMBER(22,4) | Payment received for a non-delinquent account between last payment due date and statement generation date. |

This procedure returning summary by account statement.

* + 1. LOYALTY\_POINTS\_SUM

Input parameters:

i\_inst\_id – Institute identifier;

i\_account\_number – Number of account (not mandatory);

i\_invoice\_id – Invoice identifier (not mandatory);

One of i\_account\_number or i\_invoice\_id should be filled.

Output parameters:

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Number of account |
| CURRENT\_POINTS | NUMBER(22,4) | Total available points |
| EARNED\_POINTS | NUMBER(22,4) | Points earned for transaction where authorization is done in Billing cycle. period and settlement received before statement generation date |
| EXPIRE\_POINTS | NUMBER(22,4) | Points expiring at the end of current year |

This procedure returning summary of loyalty points by account statement.

* + 1. CREDIT\_SERVICE\_LIMIT

Input parameters:

i\_inst\_id – Institute identifier;

i\_account\_number – Number of account (not mandatory);

i\_invoice\_id – Invoice identifier (not mandatory);

One of i\_account\_number or i\_invoice\_id should be filled.

Output parameters:

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Number of account |
| TOTAL\_CREDIT\_LIMIT | NUMBER(22,4) | Credit Limit at account level |
| TOTAL\_CASH\_LIMIT | NUMBER(22,4) | Total cash limit at account level |

This procedure returning summary of limits of credit service.

* + 1. ACCOUNT\_CARD

Input parameters:

i\_inst\_id – Institute identifier;

i\_account\_number – Number of account (not mandatory);

i\_invoice\_id – Invoice identifier (not mandatory);

One of i\_account\_number or i\_invoice\_id should be filled.

Output parameters:

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Number of account |
| CARD\_TYPE | VARCHAR2(8) | Category of card |
| CARD\_NUMBER | VARCHAR2(24) | Masked Card Number |
| SAVING\_ACCOUNT\_NUMBER | VARCHAR2(32) | CBS savings account number linked to Credit Account in SV |
| PRODUCT\_NAME | VARCHAR2(200) | Card Product name |

This procedure returning information of cards by account statement.

* + 1. DUE\_AMOUNTS

Input parameters:

i\_inst\_id – Institute identifier;

i\_account\_number – Number of account (not mandatory);

i\_invoice\_id – Invoice identifier (not mandatory);

One of i\_account\_number or i\_invoice\_id should be filled.

Output parameters:

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Number of account |
| DESCRIPTION | VARCHAR2(100) | Transaction description for manual or generated transaction in system |
| CURRENCY | VARCHAR2(3) | Currency of account |
| TRANSACTION\_AMOUNT | NUMBER(22,4) | Destination Amount |
| BILLING\_PRINCIPAL\_AMOUNT | NUMBER(22,4) | Transaction Billing amount posted on account |
| INTEREST\_FEE | NUMBER(22,4) | Interest of Transaction + Transaction fee (online) |

This procedure returning summary of transactions for accounts divided by delinquent and non-delinquent.

* + 1. DUE\_AMOUNTS

Input parameters:

i\_inst\_id – Institute identifier;

i\_account\_number – Number of account (not mandatory);

i\_invoice\_id – Invoice identifier (not mandatory);

i\_currency – National Currency;

i\_rate\_type – Type of rate for converting amounts;

One of i\_account\_number or i\_invoice\_id should be filled.

Output parameters:

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Number of account |
| TRANSACTION\_DATE | DATE | Transaction Date |
| ACQUIRE\_DATE | DATE | Settlement date (Transaction Posting Date) |
| MERCHANT\_NAME | VARCHAR2(200) | Merchant Name |
| CURRENCY | VARCHAR2(3) | Transaction Currency Code |
| TRANSACTION\_AMOUNT | NUMBER(22,4) | Transaction Amount |
| DISCOUNT | NUMBER(22,4) | Discount on Transaction |
| OVERSEAS | NUMBER(22,4) | Base Currency (USD) amount from VISA for foreign currency transaction |
| BILLING\_PRINCIPAL\_AMOUNT | NUMBER(22,4) | Transaction Billing amount posted on account |
| INTEREST\_FEE | NUMBER(22,4) | Interest of Transaction + Transaction fee (online) |
| EXCHANGE\_RATE | NUMBER | Currency exchange rate between (USD) base currency and destination currency |
| POINT\_NAME | VARCHAR2(200) | Loyalty Program name |
| EARNED\_POINT | NUMBER(22,4) | Points earned on transaction |
| DPP\_PERIOD | NUMBER(4) | Number of total installments for DPP |
| DPP\_COUNT | NUMBER(4) | Installment Billed to customer |
| REMAINING\_BALANCE | NUMBER(22,4) | Remaining principal balance on DPP |
| CARD\_NUMBER | VARCHAR2(24) | Masked Card Number |
| CARD\_ID | NUMBER(12) | Card identifier |

This procedure returning transactions list grouped by account’s cards.

* + 1. GET\_CUSTOMER\_CRD\_INVOICE\_INFO

Input parameters:

i\_customer\_id – Identifier of the customer (mandatory);

i\_agent\_id – Agent identifier (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| INVOICE\_ID | NUMBER(12) | Identifier of the invoice |
| CREDIT\_ACCOUNT\_NUMBER | VARCHAR2(32) | Number of account |
| TOTAL\_AMOUNT\_DUE | NUMBER(22, 4) | Total amount due |
| MIN\_AMOUNT\_DUE | NUMBER(22, 4) | Minimum amount due |
| OVERDUE\_BALANCE | NUMBER(22, 4) | Amount of overdue balance |
| PAYMENT\_AMOUNT | NUMBER(22, 4) | Payments amount made in last period |
| DUE\_DATE | DATE | Due date |
| EXPENSE\_AMOUNT | NUMBER(22, 4) | Debit operations amount made in last period |

This procedure returning main parameters of the invoices for customer credit accounts. Fields which returning are listed higher.

* + 1. GET\_CRD\_INVOICE\_PAYMENTS\_INFO

Input parameters:

i\_invoice\_id– Identifier of the invoice (mandatory);

i\_agent\_id – Agent identifier (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| OPER\_DATE | DATE | Date of the operation |
| AMOUNT | NUMBER(22, 4) | Total amount of the payment |

This procedure returning parameters of the payments for invoice of the customer credit account.

1. Visa
   1. Overview

API for getting specific Visa data. Procedures, functions, data structures are described below.

* 1. References

Procedures are in the following packages:

* vis\_api\_transaction\_pkg
  1. List of PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| vis\_api\_transaction\_pkg | get\_unheld\_transactions | procedure | Returns ref\_cursor |
| vis\_api\_transaction\_pkg | get\_deducted\_transactions | procedure | Returns ref\_cursor |
| vis\_api\_transaction\_pkg | get\_transactions\_without\_sttl | procedure | Returns ref\_cursor |
| vis\_api\_transaction\_pkg | get\_transactions\_without\_auth | procedure | Returns ref\_cursor |

* + 1. GET\_UNHELD\_TRANSACTIONS

Input parameters:

i\_inst\_id – institution identifier;

i\_agent\_id – agent indentifier;

i\_start\_date – start date for transaction entries;

i\_end\_date – end date for transaction entries.

Output parameters:

o\_ref\_cursor — result cursor of type ref\_cursor with accounting entries of transaction type TRNT0106 (Cancellation of hold on client account) for Visa financial messages. Only cards with feature CFCHDEBT (Debit) in reference table NET\_CARD\_TYPE\_FEATURE are taken into account.

Result cursor contains the following fields.

| Field | Type | Description |
| --- | --- | --- |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Account number |
| CARD\_NUMBER | VARCHAR2(24) | Card number from Visa fin. message |
| AUTH\_CODE | DATE | Authorization code from Visa fin. message |
| OPER\_DATE | DATE | Operation date from Visa fin.message |
| AGENT\_NUMBER | VARCHAR2(200) | Agent number associated with account, if this value is not defined for account’s agent then its ID is returned |
| TRANSACTION\_CURRENCY | VARCHAR2(3) | Currency code of accounting entry |
| TRANSACTION\_AMOUNT | NUMBER(22, 4) | Amount of accounting entry |
| CUSTOMER\_NUMBER | VARCHAR2(200) | Customer number |

The cursor is ordered by account number, card number, operation date and authorization code.

* + 1. GET\_DEDUCTED\_TRANSACTIONS

Input parameters:

i\_inst\_id – institution identifier;

i\_agent\_id – agent indentifier;

i\_start\_date – start date for transaction entries;

i\_end\_date – end date for transaction entries.

Output parameters:

o\_ref\_cursor — result cursor of type ref\_cursor with accounting entries from balances of types BLTP0001 (Ledger) and BLTP0003 (Fees) for Visa financial messages. Only cards with feature CFCHDEBT (Debit) in reference table NET\_CARD\_TYPE\_FEATURE are taken into account.

Result cursor contains the following fields.

| Field | Type | Description |
| --- | --- | --- |
| CUSTOMER\_NUMBER | VARCHAR2(200) | Customer number |
| AGENT\_NUMBER | VARCHAR2(200) | Agent number associated with account, if this value is not defined for account’s agent then its ID is returned |
| ACCOUNT\_NUMBER | VARCHAR2(32) | Account number |
| TRANSACTION\_AMOUNT | NUMBER(22, 4) | Amount of accounting entries |

All accounting entries are grouped by the following fields: customer number, agent number and account number. Transaction amount contains total amount of all accounting entries after this grouping.

The cursor is ordered by customer number and account number.

* + 1. GET\_TRANSACTIONS\_WITHOUT\_STTL

Input parameters:

i\_inst\_id – institution identifier;

i\_agent\_id – agent indentifier;

i\_start\_date – start date for transaction entries;

i\_end\_date – end date for transaction entries.

Output parameters:

o\_ref\_cursor — result cursor of type ref\_cursor with operations from SVFE without settlement from Visa. These operations have message type MSGTAUTH (Authorization), MSGTPREU (Preauthorization), MSGTCMPL (Authorization completion) and matching status MTST0200 (Require matching) or MTST0600 (Partially matched).

Result cursor contains the following fields.

| Field | Type | Description |
| --- | --- | --- |
| CARD\_NUMBER | VARCHAR2(24) | Card number from operation |
| ARN | VARCHAR2(23) | Acquirer reference number. This field forms the same way as for incoming Visa clearing |
| OPER\_DATE | DATE | Operation date |
| OPER\_AMOUNT | NUMBER(22, 4) | Operation amount |
| OPER\_CURRENCY | VARCHAR2(3) | Operation currency code |
| MERCHANT\_NAME | VARCHAR2(200) | Operation merchant name |
| MERCHANT\_CITY | VARCHAR2(200) | Operation merchant city |
| AUTH\_CODE | DATE | Operation authorization code |
| POS\_ENTRY\_MODE | VARCHAR2(3) | POS entry mode value that is formed the same way as for incoming Visa clearing, based on Card data input mode flag |

The cursor is ordered by card number, operation date and merchant name.

* + 1. GET\_TRANSACTIONS\_WITHOUT\_AUTH

Input parameters:

i\_inst\_id – institution identifier;

i\_agent\_id – agent indentifier;

i\_start\_date – start date for transaction entries;

i\_end\_date – end date for transaction entries.

Output parameters:

o\_ref\_cursor — result cursor of type ref\_cursor with operations without corresponding authorizations from SVFE. These operations have message type MSGTPRES (Presentment), MSGTPAMC (Partial amount, multi-clearing) or MSGTPACC (Partial amount , completion) and matching status MTST0200 (Require matching) or MTST0600 (Partially matched).

Result cursor contains the following fields.

| Field | Type | Description |
| --- | --- | --- |
| CARD\_NUMBER | VARCHAR2(24) | Card number from Visa fin. message |
| ARN | VARCHAR2(23) | Acquirer reference number from Visa fin. message |
| OPER\_DATE | DATE | Operation date from Visa fin. message |
| STTL\_AMOUNT | NUMBER(22, 4) | Settlement amount from Visa fin. message |
| STTL\_CURRENCY | VARCHAR2(3) | Settlement currency code from Visa fin. message |
| OPER\_AMOUNT | NUMBER(22, 4) | Operation amount from Visa fin. message |
| OPER\_CURRENCY | VARCHAR2(3) | Operation currency code from Visa fin. message |
| MERCHANT\_NAME | VARCHAR2(200) | Merchant name from Visa fin. message |
| MERCHANT\_CITY | VARCHAR2(200) | Merchant city from Visa fin. message |
| AUTH\_CODE | DATE | Authorization code from Visa fin. message |
| POS\_ENTRY\_MODE | VARCHAR2(3) | POS entry mode value from Visa fin. message |

The cursor is ordered by card number, operation date and merchant name.

1. Scoring
   1. Overview
   2. Referencies
   3. List of PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| scr\_api\_external\_pkg | add\_bucket | procedure | Returns ref\_cursor |
| scr\_api\_external\_pkg | get\_scoring\_data | procedure | Returns ref\_cursor |

* + 1. ADD\_BUCKET

Input parameters 1:

i\_account\_id - Account identifier;

i\_customer\_id – customer identifier;

i\_revised\_bucket - VARCHAR2(2) - Manually bucket to update to SV;

i\_eff\_date - The date manually bucket is effective;

i\_expir\_date - The date manually bucket is expire;

i\_valid\_period - NUMBER (3)- Period that manually bucket is effective;

i\_reason - VARCHAR2(128) - Reason to adjust Bucket;

i\_user\_id - VARCHAR2(32) - Person in charge of revising bucket.

Overloaded procedure

Input parameters 2:

i\_bucket\_tab – collection of bucket records.

The procedure inserts revised bucket data into scr\_bucket table via scr\_bucket\_vw.

* + 1. GET\_SCORING\_DATA

Input parameters:

i\_inst\_id – institution identifier;

i\_agent\_id – agent indentifier;

i\_customer\_id – customer identifier;

i\_account\_id – account identifier;

i\_start\_date – start date for scoring data;

i\_end\_date – end date for scoring data.

Output parameters:

o\_ref\_cursor — result cursor of type ref\_cursor with common data from SVBO tables.

Result cursor contains the following fields.

| Field | Type | Description |
| --- | --- | --- |
| CUSTOMER\_ID | VARCHAR2(32) | Customer number in SV |
| ACCOUNT\_ID | VARCHAR2(32) | Account number in SV |
| CARD\_MASK | VARCHAR2(24) | Masked Card number in SV |
| CATEGORY | VARCHAR2(8) | Card category |
| CARD\_STATUS | VARCHAR2(8) | Current status of card |
| CARD\_LIMIT | NUMBER(22,4) | Spending limit of card |
| BILLING\_DATE | DATE | For current cycle |
| DUE\_DATE | DATE | For current cycle DUE DATE 2 |
| MAD\_CYCLE | NUMBER(22,4) | For current cycle MINIMUM AMOUNT DUE 2 |
| AVAILABLE\_LIMIT | NUMBER(22,4) | Available credit limit (open-to-buy) |
| BALANCE\_TYPE | VARCHAR2 (32) | Balance type |
| BALANCE\_AMOUNT | NUMBER (16) | Balance amount |
| COUNT\_ATM | NUMBER (12) | Number of ATM withdraw transaction. Occur in data date |
| COUNT\_CASH | NUMBER (12) | Number of Cash advance at counter. Occur in data date |
| TRANSACTION\_COUNT | NUMBER (12) | All kind of transaction occur in data date |
| AMOUNT\_ATM | NUMBER (22,4) | Amount of ATM withdraw transaction. Occur in data date |
| AMOUNT\_CASH | NUMBER (22,4) | Amount of Cash advance at counter. Occur in data date |
| TRANSACTION\_AMOUNT | NUMBER (22,4) | Total daily Amount of transaction. All kind of transaction occur in data date |
| TOTAL\_DAILY\_REPAYMENT | NUMBER (22,4) | Total daily repayment. Occur in data date |
| TOTALCYCLE\_REPAYMENT | NUMBER (22,4) | Total repayment in cycle. Accumulated amount for current cycle |
| DPD | NUMBER (4,0) | Current DPD |
| BUCKET | VARCHAR2 (2) | Current bucket |
| REVISED\_BUCKET | VARCHAR2 (2) | Current revised bucket |
| EFF\_DATE | DATE | Effective date. Revised bucket incoming file from scoring |
| EXPIR\_DATE | DATE | Expired date. Revised bucket incoming file from scoring |
| VALID\_PERIOD | NUMBER (3) | Valid period. Revised bucket incoming file from scoring |
| REASON | VARCHAR2 (128) | Reason. Revised bucket incoming file from scoring |
| HIGH\_REVISED\_BUCKET\_CYCLE | VARCHAR2 (2) | Highest revised bucket of current cycle. System to store the highest revised value for a cycle |
| HIGH\_REVISED\_BUCKET\_CYCLE\_3 | VARCHAR2 (2) | Highest revised bucket of 03 last cycle. System to store the highest revised value for a last 3 cycles |
| HIGH\_REVISED\_BUCKET\_CYCLE\_6 | VARCHAR2 (2) | Highest revised bucket of 06 last cycle. System to store the highest revised value for a last 6 cycles |
| HIGH\_DPD\_CYCLE | NUMBER (4) | Highest DPD of current cycle. System should store highest DPD of a cycle |
| AMOUNT\_CASH\_CYCLE | NUMBER (22,4) | Total cash withdrawal in a cycle |
| TRANSACTION\_AMOUNT\_CYCLE | NUMBER (22,4) | Total amount of debits in current cycle |
| AVG\_CASH\_CYCLE | NUMBER (22,4) | total cash withdrawn in cycle/ total number of transaction in cycle |
| DAILY\_AVG\_CASH\_CYCLE | NUMBER (22,4) | Total cash withdrawal in a cycle/number of days in cycle till date |
| TOTAL\_CASH\_LIFE | NUMBER (22,4) | Card usage. Total cash withdraw Life to date (till date since account open) |
| TRANSACTION\_COUNT\_LIFE | NUMBER (12) | Accumulated transaction. Number of transaction so far from open date till date |
| AVG\_CASH\_LIFE | NUMBER (22,4) | Avg withdraw. Total cash withdrawal till date/ Accumulated transation |
| DAILY\_AVG\_CASH\_LIFE | NUMBER (22,4) | Daily Avg usage. Card usage/ (Date of report generation - First withdraw date) |
| MONTHLY\_AVG\_CASH\_LIFE | NUMBER (22,4) | Monthly Avg usage. Card usage/ Existing month from first withdraw date |
| TEMPORARY\_CREDIT\_LIMIT | NUMBER (22,4) | Temporary credit limit |
| EFF\_DATE\_TEMP\_CREDIT\_LIMIT | DATE | Effective date of Temporary credit limit |
| EXPIR\_DATE\_TEMP\_CREDIT\_LIMIT | DATE | Expired date of Temporary credit limit |
| AMOUNT\_CASH\_TEMP\_PERIOD | NUMBER (22,4) | Card usage in Temporary credit limit period. Total cash withdraw while Temporary credit limit valid |
| OVERDUE\_INTEREST\_AMOUNT | NUMBER (22,4) | Accumulated overdue interest amount as of reporting date |
| INDUE\_INTEREST\_AMOUNT | NUMBER (22,4) | Accumulated In-due interest amount as of reporting date |

1. Events
   1. Overview

Following methods provide API for work with events. Procedures, functions, data structures are described below.

* 1. References

Procedures are placed in packages:

evt\_api\_external\_pkg

* 1. List of procedures

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| evt\_api\_external\_pkg | get\_events | proc | Return count of records and ref\_cursor |

* + 1. GET\_EVENTS

Input parameters:

i\_inst\_id – institution identifier (optional, by default user institution);

i\_procedure\_name – procedure name (mandatory);

i\_start\_date – start date (optional, by default current date 00:00:00);

i\_end\_date – end date (optional, by default current date 23:59:59)

i\_entity\_type – entiy type (optional);

i\_object\_id – object identifier (optional);

i\_split\_hash – object split hash (optional, should by find by entity and object if it is empty);

i\_status – event status (optional, by default EVST0001).

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| ID | NUMBER(16) | Unique identifier |
| EVENT\_TYPE | VARCHAR2(8) | Event type |
| PROCEDURE\_NAME | VARCHAR2(200) | Procedure name |
| ENTITY\_TYPE | VARCHAR2(8) | Entity type |
| OBJECT\_ID | NUMBER(16) | Entity object identifier |
| EFF\_DATE | DATE | Effective date |
| INST\_ID | NUMBER(4) | Institution identifier |
| SPLIT\_HASH | NUMBER(4) | Number of account |
| STATUS | VARCHAR2(8) | ID of the term account agent |

The procedure returns records count and events cursor by incoming parameters.

1. OPERATIONS
   1. OVERVIEW

Following methods provide API for work with operations objects and theirs data. Procedures, functions, data structures are described below.

* 1. REFERENCES

Procedures are placed in packages:

opr\_api\_external\_pkg

* 1. LIST OF PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| opr\_api\_external\_pkg | get\_operations\_data | proc | Return count of record and ref\_cursor with operation detail information |
| opr\_api\_external\_pkg | get\_oper\_participants\_data | proc | Returns count of records and ref\_cursor with operation participants data |
| opr\_api\_external\_pkg | get\_aggr\_oper\_transact\_data | proc | Return count of records and ref\_cursor with operations and linked transactions aggregation data |

* + 1. GET\_OPERATIONS\_DATA

Input parameters:

i\_inst\_id – Identifier of the institution (mandatory);

i\_participant\_type - The type of the operation participant (dictionary article) there is a same institution for the selection of operations (not mandatory);

i\_start\_date - Date begin (for default – current date on day begin) (mandatory);

i\_end\_date - Date end (for default – current date on day end) (mandatory);

i\_oper\_currency – Currency of the operations (not mandatory);

i\_object\_tab – Table of objects which use in select of transactions, with defined structure: level\_type (entity type of level for define of link between current object and transactions), entity\_type (entity type of object), object\_id (table of the objects identifier) (not mandatory);

i\_array\_operations\_type\_id - Identifier of operation types array (not mandatory);

i\_array\_ oper\_statuses\_id - Identifier of operation statuses array (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| oper\_num | number(8) | Number of unloading row |
| oper\_id | number(16) | opr\_operation.id |
| oper\_type | varchar2(8) | opr\_operation.oper\_type |
| oper\_reason | varchar2(8) | opr\_operation.oper\_reason |
| is\_reversal | number(1) | opr\_operation.is\_reversal |
| msg\_type | varchar2(8) | opr\_operation.msg\_type |
| sttl\_type | varchar2(8) | opr\_operation.sttl\_type |
| oper\_amount | number(22,4) | opr\_operation.oper\_amount |
| oper\_currency | varchar2(3) | opr\_operation.oper\_currency |
| oper\_date | date | opr\_operation.oper\_date |
| host\_date | date | opr\_operation.host\_date |
| unhold\_date | date | opr\_operation.unhold\_date |
| oper\_sttl\_date | date | opr\_operation.sttl\_date |
| oper\_status | varchar2(8) | opr\_operation.status |
| oper\_status\_reason | varchar2(8) | opr\_operation.status\_reason |
| terminal\_type | varchar2(8) | opr\_operation.terminal\_type |
| acq\_inst\_bin | varchar2(24) | opr\_operation.acq\_inst\_bin |
| forw\_inst\_bin | varchar2(24) | opr\_operation.forw\_inst\_bin |
| merchant\_number | varchar2(15) | opr\_operation.merchant\_number |
| terminal\_number | varchar2(16) | opr\_operation.terminal\_number |
| merchant\_name | varchar2(200) | opr\_operation.merchant\_name |
| merchant\_street | varchar2(200) | opr\_operation.merchant\_street |
| merchant\_city | varchar2(200) | opr\_operation.merchant\_city |
| merchant\_region | varchar2(11) | opr\_operation.merchant\_region |
| merchant\_country | varchar2(3) | opr\_operation.merchant\_country |
| merchant\_postcode | varchar2(10) | opr\_operation.merchant\_postcode |
| mcc | varchar2(4) | opr\_operation.mcc |
| originator\_refnum | varchar2(200) | opr\_operation.originator\_refnum |
| network\_refnum | varchar2(200) | opr\_operation.network\_refnum |
| match\_status | varchar2(8) | opr\_operation.match\_status |
| dispute\_id | number(16) | opr\_operation.dispute\_id |
| payment\_order\_id | number(16) | opr\_operation.payment\_order\_id |
| payment\_host\_id | number(4) | opr\_operation.payment\_host\_id |
| fee\_amount | number(22,4) | opr\_operation.fee\_amount |
| fee\_currency | varchar2(3) | opr\_operation.fee\_currency |
| resp\_code | varchar2(8) | aut\_auth.resp\_code |
| proc\_type | varchar2(8) | aut\_auth.proc\_type |
| proc\_mode | varchar2(8) | aut\_auth.proc\_mode |
| bin\_amount | number(22,4) | aut\_auth.bin\_amount |
| bin\_currency | varchar2(3) | aut\_auth.bin\_currency |
| bin\_cnvt\_rate | number(22,4) | aut\_auth.bin\_cnvt\_rate |
| network\_amount | number(22,4) | aut\_auth.network\_amount |
| network\_currency | varchar2(3) | aut\_auth.network\_currency |
| network\_cnvt\_date | date | aut\_auth.network\_cnvt\_date |
| network\_cnvt\_rate | number(22,4) | aut\_auth.network\_cnvt\_rate |
| account\_cnvt\_rate | number(22,4) | aut\_auth.account\_cnvt\_rate |
| transaction\_id | varchar2(200) | aut\_auth.transaction\_id |
| system\_trace\_audit\_number | varchar2(6) | aut\_auth.system\_trace\_audit\_number |
| external\_auth\_id | varchar2(30) | aut\_auth.external\_auth\_id |
| external\_orig\_id | varchar2(30) | aut\_auth.external\_orig\_id |
| agent\_unique\_id | varchar2(30) | aut\_auth.agent\_unique\_id |
| trace\_number | varchar2(30) | aut\_auth.trace\_number |
| orig\_msg\_type | varchar2(8) | opr\_operation.msg\_type (original) |
| orig\_sttl\_type | varchar2(8) | opr\_operation.sttl\_type (original) |
| orig\_oper\_amount | number(22,4) | opr\_operation.oper\_amount (original) |
| orig\_oper\_currency | varchar2(3) | opr\_operation.oper\_currency (original) |
| orig\_oper\_date | date | opr\_operation.oper\_date (original) |
| orig\_host\_date | date | opr\_operation.host\_date (original) |
| orig\_unhold\_date | date | opr\_operation.unhold\_date (original) |
| orig\_terminal\_type | varchar2(8) | opr\_operation.terminal\_type (original) |
| orig\_acq\_inst\_bin | varchar2(24) | opr\_operation.acq\_inst\_bin (original) |
| orig\_forw\_inst\_bin | varchar2(24) | opr\_operation.forw\_inst\_bin (original) |
| orig\_originator\_refnum | varchar2(200) | opr\_operation.originator\_refnum (original) |
| orig\_network\_refnum | varchar2(200) | opr\_operation.network\_refnum (original) |
| orig\_bin\_amount | number(22,4) | aut\_auth.bin\_amount (original) |
| orig\_bin\_currency | varchar2(3) | aut\_auth.bin\_currency (original) |
| orig\_bin\_cnvt\_rate | number(22,4) | aut\_auth.bin\_cnvt\_rate (original) |
| orig\_network\_amount | number(22,4) | aut\_auth.network\_amount (original) |
| orig\_network\_currency | varchar2(3) | aut\_auth.network\_currency (original) |
| orig\_network\_cnvt\_date | date | aut\_auth.network\_cnvt\_date (original) |
| orig\_network\_cnvt\_rate | number(22,4) | aut\_auth.network\_cnvt\_rate (original) |
| orig\_account\_cnvt\_rate | number(22,4) | aut\_auth.account\_cnvt\_rate (original) |
| orig\_transaction\_id | varchar2(200) | aut\_auth.transaction\_id (original) |
| orig\_system\_trace\_audit\_number | varchar2(6) | aut\_auth.system\_trace\_audit\_number (original) |
| orig\_external\_auth\_id | varchar2(30) | aut\_auth.external\_auth\_id (original) |
| orig\_external\_orig\_id | varchar2(30) | aut\_auth.external\_orig\_id (original) |
| orig\_agent\_unique\_id | varchar2(30) | aut\_auth.agent\_unique\_id (original) |
| orig\_trace\_number | varchar2(30) | aut\_auth.trace\_number (original) |

This procedure returning main detail parameters of the operations. Fields which returning are listed higher.

* + 1. GET\_OPER\_PARTICIPANTS\_DATA

Input parameters:

i\_inst\_id – Identifier of the institution (mandatory);

i\_oper\_id – Identifier of the operation (mandatory);

i\_array\_oper\_participant\_type - Identifier of operation participant types array (not mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| oper\_id | number(16) | opr\_participant.oper\_id |
| participant\_type | varchar2(8) | opr\_participant.participant\_type |
| client\_id\_type | varchar2(8) | opr\_participant.client\_id\_type |
| client\_id\_value | varchar2(200) | opr\_participant.client\_id\_value |
| inst\_id | number(4) | opr\_participant.inst\_id |
| network\_id | number(4) | opr\_participant.network\_id |
| card\_inst\_id | number(4) | opr\_participant.card\_inst\_id |
| card\_network\_id | number(4) | opr\_participant.card\_network\_id |
| card\_id | number(12) | opr\_participant.card\_id |
| card\_instance\_id | number(12) | opr\_participant.card\_instance\_id |
| card\_type\_id | number(4) | opr\_participant.card\_type\_id |
| card\_number | varchar2(24) | opr\_card.card\_number (after decode) |
| card\_mask | varchar2(24) | opr\_participant.card\_mask |
| card\_hash | number(12) | opr\_participant.card\_hash |
| card\_seq\_number | number(4) | opr\_participant.card\_seq\_number |
| card\_expir\_date | date | opr\_participant.card\_expir\_date |
| card\_service\_code | varchar2(3) | opr\_participant.card\_service\_code |
| card\_country | varchar2(3) | opr\_participant.card\_country |
| customer\_id | number(12) | opr\_participant.customer\_id |
| account\_id | number(12) | opr\_participant.account\_id |
| account\_type | varchar2(8) | opr\_participant.account\_type |
| account\_number | varchar2(32) | opr\_participant.account\_number |
| account\_amount | number(22,4) | opr\_participant.account\_amount |
| account\_currency | varchar2(3) | opr\_participant.account\_currency |
| auth\_code | varchar2(6) | opr\_participant.auth\_code |
| merchant\_id | number(8) | opr\_participant.merchant\_id |
| terminal\_id | number(8) | opr\_participant.terminal\_id |
| split\_hash | number(4) | opr\_participant.split\_hash |
| debit\_entry\_impact | number(1) | attribute what current participant is debit impact in entry transaction |
| credit\_entry\_impact | number(1) | attribute what current participant is credit impact in entry transaction |

This procedure returning main detail parameters of the operation participants. Fields which returning are listed higher.

* + 1. GET\_AGGR\_OPER\_TRANSACT\_DATA

Input parameters:

i\_inst\_id – Identifier of the institution (mandatory);

i\_start\_date - Date begin (for default – current date on day begin) (mandatory);

i\_end\_date - Date end (for default – current date on day end) (mandatory);

i\_object\_tab – Table of objects which use in select of transactions, with defined structure: level\_type (entity type of level for define of link between current object and transactions), entity\_type (entity type of object), object\_id (table of the objects identifier) (not mandatory);

i\_oper\_currency – Currency of the operations (not mandatory);

i\_array\_oper\_participant\_type - Identifier of operation participant types array (not mandatory);

i\_array\_operations\_type\_id - Identifier of operation types array (not mandatory);

i\_array\_ oper\_statuses\_id - Identifier of operation statuses array (not mandatory);

i\_aggr\_operations\_type – attribute, aggregation under operation type field (mandatory);

i\_aggr\_opr\_participant – attribute, aggregation under participant type field (mandatory);

i\_aggr\_terminal\_type – attribute, aggregation under terminal type field (mandatory);

i\_aggr\_balance\_impact – attribute, aggregation under acc\_entry.balance\_impact field (mandatory);

i\_mask\_error – Include or exclude raise (not mandatory);

Output parameters:

o\_row\_count - Number of rows returned;

o\_ref\_cursor - Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| oper\_type | varchar2(8) | opr\_operation.oper\_type |
| participant\_type | varchar2(8) | opr\_participant.participant\_type |
| terminal\_type | varchar2(8) | opr\_operation.terminal\_type |
| balance\_impact | number(1) | acc\_entry.balance\_impact |
| entry\_currency | varchar2(3) | acc\_entry.currency |
| amount | number(4) | acc\_entry.amount (aggregation value) |

This procedure returning main aggregation parameters of the operations, participants, entries. Fields which returning are listed higher.

* + 1. GET\_OPR\_ADDITIONAL\_AMOUNT

Input parameters:

i\_oper\_id – Identifier of operation (mandatory)

i\_amount\_type – Amount type

Output parameters:

o\_ref\_cursor - Result cursor

Resultset of the cursor matches sections “Additional amount” of clearing file structure in svxp\_cbs\_eng.docx

To fetch records of this cursor type opr\_api\_external\_pkg.t\_opr\_add\_amount\_tab should be used.

* + 1. GET\_OPR\_PARTICIPANTS

Input parameters:

i\_oper\_id – Identifier of operation (mandatory)

i\_participant\_type – participant type

Output parameters:

o\_ref\_cursor - Result cursor

Resultset of the cursor matches section “Participant” of clearing file structure in svxp\_cbs\_eng.docx

To fetch records of this cursor type opr\_api\_external\_pkg.t\_opr\_participant\_tab should be used.

* + 1. GET\_AUTH\_TAG\_VALUES

Input parameters:

i\_oper\_id – Identifier of operation (mandatory)

i\_tag\_reference – tag reference

Output parameters:

o\_ref\_cursor - Result cursor

Resultset of the cursor matches section “auth\_tag” of clearing file structure in svxp\_cbs\_eng.docx

To fetch records of this cursor type aup\_api\_external\_pkg.t\_auth\_tag\_tab should be used.

* + 1. GET\_OPR\_CLEARING\_DATA

Input parameters:

i\_oper\_id – Identifier of operation (mandatory)

Output parameters:

o\_ipm\_data opr\_api\_external\_pkg.t\_ipm\_data\_rec

o\_baseii\_data opr\_api\_external\_pkg.t\_baseii\_data\_rec

Type description of t\_ipm\_data\_rec, t\_baseii\_data\_rec matches sections “ipm\_data”, “baseII\_data” of clearing file structure in svxp\_cbs\_eng.docx

* + 1. GET\_OPERATIONS

Returns operations data based on entries information

This method is

Input parameters:

i\_inst\_id – Identifier of the institution (mandatory)

i\_date\_type – Type of date used to select of transactions (mandatory)

i\_start\_date - Date begin (mandatory)

i\_end\_date - Date end (mandatory)

i\_array\_balance\_type\_id - Identifier of account balances array

i\_array\_trans\_type\_id - Identifier of transaction types array

i\_array\_settl\_type\_id - Identifier of settlement types array

i\_array\_operations\_type\_id - Identifier of operation types array

Output parameters:

o\_ref\_cursor - Result cursor

Resultset of the cursor matches sections “operation” + “auth\_data” of clearing file structure in svxp\_cbs\_eng.docx

To fetch records of this cursor type opr\_api\_external\_pkg.t\_turnover\_info\_tab should be used.

1. Payment orders
   1. OVERVIEW

Following methods provide API for work with payment orders objects and theirs data. Procedures, functions, data structures are described below.

* 1. REFERENCES

Procedures are placed in packages:

pmo\_api\_order\_pkg

* 1. LIST OF PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| pmo\_api\_order\_pkg | get\_order\_list | proc | Return ref\_cursor with a list of payment orders information |
| pmo\_api\_order\_pkg | get\_order\_list\_count | proc | Return count of record of payment orders information |
| pmo\_api\_order\_pkg | get\_order\_evt\_list | proc | Returns ref\_cursor with a list of payment orders, that are associated with the specified event subscriber |
| pmo\_api\_order\_pkg | get\_order\_evt\_list\_count | proc | Return count of record of a list of payment orders, that are associated with the specified event subscriber |

* + 1. GET\_ORDER\_LIST

Input parameters:

i\_inst\_id – Identifier of the institution (mandatory);

i\_purpose\_id – Identifier of the payment purpose (mandatory);

i\_status – Status of the payment order ( mandatory);

i\_service\_provider\_id – Identifier of the Service Provider ( not mandatory);

o\_order\_list – Result cursor;

| Field | Type | Description |
| --- | --- | --- |
| id | number(16) | pmo\_order.id |
| split\_hash | number(4) | pmo\_order.split\_hash |
| customer\_id | number(12) | pmo\_order.customer\_id |
| customer\_number | varchar2(200) | prd\_customer.customer\_number |
| Amount | number(22,4) | pmo\_order.amount |
| Currency | varchar2(3) | pmo\_order.currency |
| inst\_id | number(4) | pmo\_order.inst\_id |
| event\_date | date | pmo\_order.event\_date |
| entity\_type | varchar2(8) | pmo\_order.entity\_type |
| object\_id | number(16) | pmo\_order.object\_id |
| purpose\_id | number(8) | pmo\_order.purpose\_id |

* + 1. GET\_ORDER\_LIST\_COUNT

i\_inst\_id – Identifier of the institution (mandatory);

i\_purpose\_id – Identifier of the payment purpose (mandatory);

i\_status – Status of the payment order ( mandatory);

i\_service\_provider\_id – Identifier of the Service Provider ( not mandatory);

o\_order\_list\_count – Number of rows returned by result cursor of procedure GET\_ORDER\_LIST with same parameter values

* + 1. GET\_ORDER\_EVT\_LIST

inst\_id – Identifier of the institution (mandatory);

i\_subscriber\_name – Name of subscriber (mandatory);

i\_event\_type – Event type of event objects, linked to Payment Order, EVNT dictionary (varchar2(8));

i\_purpose\_id – Identifier of the payment purpose (mandatory);

o\_order\_evt\_list – Result cursor:

| Field | Type | Description |
| --- | --- | --- |
| a.id | number(16) | evt\_event\_object.id |
| o.id as order\_id | number(16) | pmo\_order.order\_id |
| o.customer\_id | number(12) | pmo\_order.customer\_id |
| o.entity\_type | varchar2(8) | pmo\_order.entity\_type |
| o.object\_id | number(16) | pmo\_order.object\_id |
| o.purpose\_id | number(8) | pmo\_order.purpose\_id |
| o.template\_id | number(16) | pmo\_order.template\_id |
| o.amount | number(22,4) | pmo\_order.amount |
| o.currency | varchar2(3) | pmo\_order.currency |
| o.event\_date | date | pmo\_order.event\_date |
| o.status | varchar2(8) | pmo\_order.status |
| o.inst\_id | number(4) | pmo\_order.inst\_id |
| o.attempt\_count | number(4) | pmo\_order.attempt\_count |
| o.split\_hash | number(4) | pmo\_order.split\_hash |
| o.payment\_order\_number | varchar2(200) | pmo\_order.payment\_order\_number |
| c.customer\_number | varchar2(200) | prd\_customer.customer\_number |
| b.amount\_algorithm | varchar2(8) | pmo\_schedule.amount\_algorithm |

* + 1. GET\_ORDER\_EVT\_LIST\_COUNT

inst\_id – Identifier of the institution (mandatory);

i\_subscriber\_name – Name of subscriber (mandatory);

i\_event\_type – Event type of event objects, linked to Payment Order, EVNT dictionary (varchar2(8));

i\_purpose\_id – Identifier of the payment purpose (mandatory);

o\_order\_evt\_list\_count – Number of rows, returned by result cursor of procedure GET\_ORDER\_EVT\_LIST with same parameter values;

* + 1. GET\_PAYMENT\_ORDER

Input parameters:

i\_payment\_order\_id – Identifier of payment order (mandatory)

Output parameters:

o\_payment\_order pmo\_api\_external\_pkg.t\_payment\_order\_rec

o\_payment\_order\_params - Result cursor

Type description of t\_payment\_order\_rec matches section “payment\_order” of clearing file structure in svxp\_cbs\_eng.docx.

Resultset of the cursor payment\_order\_params matches sections “payment\_parameter” of clearing file structure in svxp\_cbs\_eng.docx

1. CONTRACTS
   1. Overview

Following methods provide API for work with contracts and theirs data. Procedures, functions, data structures are described below.

* 1. References

Procedures are placed in packages:

prd\_api\_external\_pkg

* 1. List of PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| prd\_api\_external\_pkg | add\_contract | proc | Add contract |

* + 1. ADD\_CONTRACT

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| o\_id | out | number(12) | Contract Id |
| o\_seqnum | out | number(4) | Contract seq.number |
| i\_product\_id | in | number(8) | Product id |
| i\_start\_date | in | date | Start date |
| i\_end\_date | in | date | End date |
| io\_contract\_number | in out | varchar2(200) | Contract number |
| i\_contract\_type | in | varchar2(8) | Contract type |
| i\_inst\_id | in | number(4) | Institution id |
| i\_agent\_id | in | number(8) | Agent id |
| i\_customer\_id | in | number(12) | Customer id |
| i\_lang | in | varchar2(8) | Language |
| i\_label | in | varchar2(200) | Label |
| i\_description | in | varchar2(2000) | Description |

Creates new contract

1. SERVICES
   1. Overview

Following methods provide API for work with services and theirs data. Procedures, functions, data structures are described below.

* 1. References

Procedures are placed in packages:

prd\_api\_external\_pkg

* 1. List of PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| prd\_api\_external\_pkg | set\_service\_object | proc | Attach object to service |

* + 1. SET\_SERVICE\_OBJECT

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| i\_service\_id | in | number(8) | Service id |
| i\_contract\_id | in | number(12) | Contract id |
| i\_entity\_type | in | varchar2(8) | Entity type |
| i\_object\_id | in | number(16) | Object id |
| i\_start\_date | in | date | Start date |
| i\_end\_date | in | date | End date |
| i\_inst\_id | in | number(4) | Institution id |

Attaches object to service

1. COMMON
   1. Overview

Following methods provide API for work with common module. Procedures, functions, data structures are described below.

* 1. References

Procedures are placed in packages:

com\_api\_external\_pkg

* 1. List of PROCEDURES

| Package | Name | Type | Description |
| --- | --- | --- | --- |
| com\_api\_external\_pkg | set\_flexible\_value | proc | Set flexible field value |
| com\_api\_external\_pkg | add\_person | proc | Add person |

* + 1. SET\_FLEXIBLE\_VALUE

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| i\_field\_name | in | varchar2(200) | i\_field\_name |
| i\_entity\_type | in | varchar2(8) | i\_entity\_type |
| i\_object\_id | in | number(16) | i\_object\_id |
| i\_seq\_number | in | number(4) | i\_seq\_number |
| i\_field\_value | in | varchar2/number/date | i\_field\_value |

Set value of flexible field. Method is overloaded for 3 data types.

* + 1. ADD\_PERSON

Input and output parameters:

| Parameter | Direction | Type | Description |
| --- | --- | --- | --- |
| io\_person\_id | in out | number(12) | person id |
| i\_lang | in | varchar2(8) | lang |
| i\_person\_title | in | varchar2(8) | person title |
| i\_first\_name | in | varchar2(200) | first name |
| i\_second\_name | in | varchar2(200) | second name |
| i\_surname | in | varchar2(200) | surname |
| i\_suffix | in | varchar2(8) | suffix |
| i\_gender | in | varchar2(8) | gender |
| i\_birthday | in | date | birthday |
| i\_place\_of\_birth | in | varchar2(200) | place of birth |
| i\_inst\_id | in | number(4) | inst id |

Creates new person

* + 1. GET\_OBJECT\_NOTES

Input parameters:

i\_object\_id – Identifier of object (mandatory)

i\_entity\_type – entity of object (mandatory)

i\_note\_type – Note type

i\_lang - language

Output parameters:

o\_ref\_cursor - Result cursor

| Field | Type | Description |
| --- | --- | --- |
| note\_type | string | Note type |
| lang | string | Attribute. Dictionary LANG. |
| note\_header | string | Header of the note |
| note\_text | String | Text of the note |

To fetch records of this cursor type com\_api\_external\_pkg.t\_notes\_tab should be used.

* + 1. GET\_OBJECT\_FLEXIBLE\_DATA

Input parameters:

i\_object\_id – Identifier of object (mandatory)

i\_entity\_type – entity of object (mandatory)

I\_field\_name – flexible field name

Output parameters:

o\_ref\_cursor - Result cursor

| Field | Type | Description |
| --- | --- | --- |
| field\_name | string | Flexible field name |
| field\_value | string | Flexible field value |

To fetch records of this cursor type com\_api\_external\_pkg.t\_flexible\_fields\_tab should be used.