API General Rules

URL Prefixes for API Contexts

The following represents the replacement value for the URL Prefix used throughout the noun definition:

API Context	URL	Header Information
User	URL = https://[domain]/user/v1/...	userContextId - this value will be required in general for APIs that exist within this context. There may be exceptions, when these exceptions exist they need to be clearly documented. • ***userContextId'*** maps to the current data model ***'cardholder.id'*** field; resolution will of this value along with other related resolution steps will be defined in the noun definition

API Action - Requirements

GET actions will default to application/json if not Content-Type is provided in the request; therefore, empty Content-Type is allowed when there is no request body.

Extra fields in valid JSON format

POST	• ignore extra field - process as if extra data was not present
PUT	• ignore extra field - process as if extra data was not present
PATCH	• ignore extra field - process as if extra data was not present • empty body in request - results in successful response with no changes to data occurring

Empty Optional field behavior for Actions

POST	• optional field is null or not present in request: field is set to persistence standard for null/empty string • required field not present in request: error 3000
PUT	• optional field is null or not present in request: field is set to persistence standard for null/empty string • Writes over any existing value • required field not present in request: error 3000
PATCH	• optional field is null or not present in request: ignore • Existing value remains in place • required field not present in request: ignore

Fields missing from an API request

- The following are equivalent resulting in error.code=3000; and are treated as equivalents
  - field is missing from the request body
  - A field is properly identified in the request body but is assigned a 'null' value

Accept-Language Header - Requirements

The API has a small amount of data this it will localize as part of the response; as opposed to most data which is either:

- Dynamic data created in the appropriate language by the admin user
- Static page data created by the application

The small amount of data that we have determined that we want the API to localize in responses is listed here:

APIs	Code field	Localized Name field
• /paymentmethods/creditdebitcards • /paymentmethods/creditdebitcards/creditDebitCardId	card.type	card.typeName
• /accountholders/{accountHolderId}/transactions • /accountholders/{accountHolderId}/accounts/{accountId}/transactions	transactionType	transactionTypeName

The Accept-Language header will be used to identify the desired language in API responses.

- the array of Accept-Language values will be prioritized in the following order:
  - qvalue assigned to language (defaults to 1.0 if not assigned)
  - position in array

The values returned in the responses for **Localized Name field** for the specified APIs will be resolved based on the following criteria; where it will exit the process upon finding the first specified matching value:

1. Look for an exact match in the Accept-Language array in order as specified above
2. Look for a partial match of the primary language code in order as specified above
3. If no matches are found provide default language value

Notes:

- If Accept-Language is not specified the default language value will be returned
- If Accept-Language is specified with invalid format the request will fail with an error message
  - This occurs if there are more than 8 chars before or after the "-" separating the primary language subtag from the extended language subtag
- Depending on implementation the Accept-Language header value may be set for your applications requests automatically, it is the application developers job to make sure this is producing the desired results.

Examples

Accept-Language Header	Supported Locales	Unsupported Locales	Response in
Accept-Language : es-MX	es	es-MX	es
Accept-Language : es-MX, fr;q=0.8	es, fr	es-MX	fr
Accept-Language : es-MX, fr-FR;q=0.8	es, fr	es-MX, fr-FR	es

Accept-Language : es-MX, fr-FR;q=0.8	es, fr-FR	es-MX	fr-FR

References

- http://www.w3.org/International/questions/qa-lang-priorities.en

API Feature Requirements

Feature Statuses

**Note: since this is currently not implemented the following examples has been removed from the noun definitions; notice the far right column*

Action	Resource	Feature
GET	/accountholders/{accountHolderId}/transactions	ACCOUNTHOLDER-TRANSACTIONS
GET	/accountholders/{accountHolderId}/accounts/{accountId}/transactions	ACCOUNTHOLDER-TRANSACTIONS
GET	/beneficiarys/{accountHolderId}/transactions	BENEFICIARY-TRANSACTIONS
GET	/beneficiarys/{accountHolderId}/accounts/{accountId}/transactions	BENEFICIARY-TRANSACTIONS

Enumeration Value	.enabled	Description	Example
0	true	ENABLED - the features is enabled	
100	false	APPLICATION-UNAVAILABLE - the feature has not been made available to this application; the feature is functional on the system	The ability to do payment transactions has not been granted to the "myCampus Kiosk" application.
101	false	APPLICATION-DISABLED - the feature has been temporarily disabled by the Application	This would be used if there was a reason to temporarily disable a feature within an application without breaking the availability linkage. For instance an application has been given the ability to update account holders but is providing bad data, so the provider has disabled this application until the problem can be resolved.
200	false	PROVIDER-DISABLED - the feature has been disabled by the provider	A selected region does not support ACH processing so all ability to provide ACH has been disabled by the provider.
300	false	ISSUER-UNAVAILABLE - the feature has not been made available to this issuer; the feature is functional on the system	This would be used if we sold feature functionality of the system to issuers; this value means that the feature is not been made available to a specific issuer.
301	false	ISSUER-DISABLED - the feature has been disabled by the Issuer	This would mean that a feature is available to the issuer such as payment transactions; but the issuer has been given the ability to temporarily disable the functionality due to a problem or business process.
500	false	BENEFICIARY-DISABLED - this feature has been disabled by the User	This would mean that the user has not been granted access to this functionality in the benefactor to beneficiary relationship.
999	false	NON-FUNCTIONAL - this feature is temporarily not functioning and has been made unavailable	The API has been released publically but is either not yet ready for use or has been deprecated.

Features Code: Values

Note: only those that are '*' can be assigned to the Benefactor by the Beneficiary - see noun 'benefactorfeatures'.

Account Holder Features	Beneficiary Holder Features
ACCOUNTHOLDER-UPDATE	
ACCOUNTHOLDER-ACCOUNTS	BENEFICIARY-ACCOUNTS
ACCOUNTHOLDER-BENEFACTORS	
ACCOUNTHOLDER-CARDS	BENEFICIARY-CARDS
ACCOUNTHOLDER-IMAGES	BENEFICIARY-IMAGES
ACCOUNTHOLDER-IMAGES-UPDATE	BENEFICIARY-IMAGES-UPDATE
ACCOUNTHOLDER-FUNDINGS	BENEFICIARY-FUNDINGS
ACCOUNTHOLDER-PAYMENTS-SCHEDULED	BENEFICIARY-PAYMENTS-SCHEDULED
ACCOUNTHOLDER-TRANSACTIONS	BENEFICIARY-TRANSACTIONS

Every feature is asserted to be '0 - ENABLED; unless able to be resolved by the following:

- OUT OF SCOPE

URL and Query Parameters - common***

URL Parameters

Data Validation Type - column

The purpose of using this is to allow for a consistent mechanism of communication around the data validation rules of a given field. The following data validation types are 'code value' representations that can be applied to each field as part of the specification.

Data Validation Type	Base DB Type	Applied Facets
Pattern_Numeric	String	Fundamental Facet(s) • valid characters = [0..9] ○ Any other characters including whitespace characters = Error

Query Parameters

Key query type & validation - column

Key query type reference	Data Validation Type	Behavior
contains_text_ci	String256	Represents the equivalent of doing a 'contains' text search where the search is: • case insensitive
enum_text_ci	String256	Represents the equivalent of doing a 'equal to' text search where the search is: • case insensitive
equals_id_string	String36ForInteger64	Data is currently persisted as an 'ID' that is a number field that may be changed to a GUID • this will only succeed if the value is an exact match
match_Date8601	Date8601	Returns the equivalent of a 24 hour period where it equates to • (greater than or equal to YYYY-MM-DDT00:00:00.000) • and (less than or equal to YYYY-MM-DDT23:59:99.999) *Note: if easier this may also be (less than YYYY-MM-DD plus 1 day)
minRange_Date8601	Date8601	min* values can be used singular or in combination with max* values; • If this is parameter is applied to a field that has precision in time this value should represent YYYY-MM-DDT00:00:00.000
maxRange_Date8601	Date8601	min* values can be used singular or in combination with max* values; • If this is parameter is applied to a field that has precision in time this value should represent YYYY-MM-DDT23:59:99.999
paging_page	Number_MinInc1MaxInc99999	Applies to paging; must be used in conjunction with query parameter key(s) or an error will occur: • paging_pageSize
paging_pageSize	Number_MinInc1MaxInc200	Applies to paging; must be used in conjunction with query parameter key(s) or an error will occur: • paging_page

Data Validation Type - for Key query type

The purpose of using this is to allow for a consistent mechanism of communication around the data validation rules of a given field. The following data validation types are 'code value' representations that can be applied to each field as part of the specification.

Data Validation Type	Base DB Type	Applied Facets
Date8601	DateTime	Fundamental Facet(s) • ISO 8061 standard date notation YYYY-MM-DD (example: 1997-07-16)
Number_MinInc1MaxInc200	Number	Fundamental Facet(s) • Minimum value = 1 • Maximum value = 200
Number_MinInc1MaxInc99999	Number	Fundamental Facet(s) • Minimum value = 1 • Maximum value = 99999

String256	String	Fundamental Facet(s) • MaxLength = 256
String36ForInteger64	Number	Fundamental Facet(s) • Data passed as string to be persisted as integer (length cannot be 0) ○ MaxLength = 36 ○ Valid Characters = [0..9] • Persisted as Integer bits = 64 ○ Inferred 'Reject' = after the valid characters are checked if the value is larger than the MaxLength the request should be rejected (not trimmed)

References:

- *The basis for this model can be found in http://www.w3.org/TR/xmlschema-2/; where this is a base data type and some facets applied to the base data type. This representation is not meant to imply physical data model but rather the data validation rules associated with each field.*
- *Identity sizes derived from: http://en.wikipedia.org/wiki/Integer_(computer_science); they can be adjusted to a larger size up for any architectural optimizations.*

Note to Development:

- *Identity and Number sizes are represented as a method to provide the 'minimal expected requirements' from the business perspective; if there is technical reason such as ease of consumption or implementation to go beyond that type it should be mentioned and the value can be changed.*

Noun Fields - common

Column Definitions

CONTEXT: User - column

The following 'code values' are used to describe the expected behavior allowed in per field in each API context; these 'code values' will appear in the field definition for the nouns of APIs:

Code Value	Field value known to context and returned in response	Allowed in API context Request	Description of behavior
No-NotApplicable	No	No - there are no API actions available in the specified context	The field update is not applicable to API context, it is listed because it may exist in other API contexts
No-NotAvailable	No	No - this field is completely unknown to the specified API context	This field is completely unknown to the specified API context
Yes-ReadOnly	Yes	Yes - this field is known in the API context	The API will not accept Create or Update requests with this field
Yes-CreateOnly	Yes	Yes - this field is known in the API context	The API will only accept Create requests with this field present
Yes-UpdateOnly	Yes	Yes - this field is known in the API context	The API will only accept Update requests with this field present
Yes-CreateAndUpdate	Yes	Yes - this field is known in the API context	The API context and is allowed change to be created or updated

Value Required - column

The column describes 'value required' expectations of the field; values may be:

Value	Description
No	The field has no value requirements the equivalent of NULL or (empty string) is allowed on add/edit
Yes	The field is not allowed to be NULL or (empty string) add/edit
Calc	There is no data persistence for this field, but rather is calculated from other information

Notes:

- *This column should not be interpreted as if the API field definition requires this field; for instance a PATCH may exclude this field but still be required by the noun definition.*

Data Validation Type - column

The purpose of using this is to allow for a consistent mechanism of communication around the data validation rules of a given field. The following data validation types are 'code value' representations that can be applied to each field as part of the specification.

Data Validation Type	Base DB Type	Applied Facets
Boolean	Boolean	Fundamental Facet(s) • Allowed values [true, false]
Currency	Number	Fundamental Facet(s) • Currency characters should be rejected and only characters related to value should be accepted • Each field will have a corresponding Currency Code dynamic reference ○ This dynamic reference field will be used to determine any validation of the decimal values *totalDigits* and *fractionDigits* validation
Currency_MinExc0	Number	Fundamental Facet(s) • Minimum value > 0 • Currency symbols should be rejected; and characters related to value should be accepted • Each field will have a corresponding Currency Code dynamic reference ○ This dynamic reference field will be used to determine any validation of the decimal values *totalDigits* and *fractionDigits* validation

Date8601	DateTime	Fundamental Facet(s) • ISO 8061 standard date notation YYYY-MM-DD (example: 1997-07-16)
DateTimeUtc	DateTime	Fundamental Facet(s) • ISO 8061 standard date notation YYYY-MM-DDThh:mm:ss.sssZ examples: ○ Accuracy to fraction of a second
Number_Enumeration_Month	Number	Business Facet(s) • Minimum value = 1 • Maximum value = 12
Number_Enumeration_Year+20	Number	Business Facet(s) • Minimum value = numeric value representing current calendar year (ex. = 2016) • Maximum value = numeric value representing current calendar year (ex. = 2036)
Number_Enumeration_MinInc0MaxInc999	Number	Fundamental Facet(s) • Minimum value = 1 • Maximum value = 999 Business Facet(s) • (see field definition for valid enumeration values)
Pattern16_Numeric	String	Fundamental Facet(s) • MaxLength = 16 • valid characters = [0..9] ○ Any other characters including whitespace characters = Error
Pattern4_NumericOrMask	String	Fundamental Facet(s) • MaxLength = 4 • valid characters = [X, 0..9] ○ Any other characters including whitespace characters = Error
Pattern6_TimeZone	String	Fundamental Facet(s) • MaxLength = 6 • valid characters = (+ or -)##:## ○ Any other characters including whitespace characters = Error
Pattern16_CreditCardNumber	String	Fundamental Facet(s) • valid characters = [0..9] ○ Any other characters including whitespace characters = Error Business Facet(s) • MinLength = 15 • MaxLength = 16 • Luhn algorithm will be applied to verify the numeric sequence; sample see: ○ http://www.freeformatter.com/credit-card-number-generator-validator.html
Pattern4_SecurityCode	String	Fundamental Facet(s) • valid characters = [0..9] ○ Any other characters including whitespace characters = Error Business Facet(s) • MinLength = 3 • MaxLength = 4
Regex20_PhoneNumber	String	Fundamental Facet(s) • MaxLength = 20 *(Requires review before additional use)* • Regex is PHONE_PATTERN = "^[\\d\\w \\-():+]{0,64}$” ○ additional validation = JavaScript that formats the entry into ###-###-####
Regex20_EmailAddrress	String	Fundamental Facet(s) • MaxLength = 255 *(Requires review before additional use)* • Regex is EMAIL_PATTERN = "/^(.+)@(.+)$/“;
String2_Enumeration	String	Fundamental Facet(s) • MaxLength = 2 • Values are case-sensitive Business Facet(s) • (see field definition for valid enumeration values)
String3_Enumeration	String	Fundamental Facet(s) • MaxLength = 3 • Values are case-sensitive Business Facet(s)

		• (see field definition for valid enumeration values)
String32_Enumeration	String	Fundamental Facet(s) • MaxLength = 32 • Values are case-sensitive Business Facet(s) • (see field definition for valid enumeration values)
String40_Enumeration	String	Fundamental Facet(s) • MaxLength = 40 • Values are case-sensitive Business Facet(s) • (see field definition for valid enumeration values)
String64_Enumeration	String	Fundamental Facet(s) • MaxLength = 64 • Values are case-sensitive Business Facet(s) • (see field definition for valid enumeration values)
String10_Trim	String	Fundamental Facet(s) • MaxLength = 10 • Trim = if the value is larger than the MaxLength that data should be trimmed to the language relevant side of the data
String20_Trim	String	Fundamental Facet(s) • MaxLength = 20 • Trim = if the value is larger than the MaxLength that data should be trimmed to the language relevant side of the data
String32_Trim	String	Fundamental Facet(s) • MaxLength = 32 • Trim = if the value is larger than the MaxLength that data should be trimmed to the language relevant side of the data
String40_Trim	String	Fundamental Facet(s) • MaxLength = 40 • Trim = if the value is larger than the MaxLength that data should be trimmed to the language relevant side of the data
String40_Trim_Nonempty	String	Fundamental Facet(s) • MaxLength = 40 • Nonempty = the corresponding field requires a non-empty string • Trim = if the value is larger than the MaxLength that data should be trimmed to the language relevant side of the data
String36ForInteger64	Number	Fundamental Facet(s) • Data passed as string to be persisted as integer (length cannot be 0) ○ MaxLength = 36 ○ Valid Characters = [0..9] • Persisted as Integer bits = 64 ○ Inferred 'Reject' = after the valid characters are checked if the value is larger than the MaxLength the request should be rejected (not trimmed)
String36ForString16_Unique_SystemId	Number	Fundamental Facet(s) • Data passed as string to be persisted as integer (length cannot be 0) ○ MaxLength = 36 ○ Valid Characters = [0..9] • Persisted as string max length 16 ○ Inferred 'Reject' = after the valid characters are checked if the value is larger than the MaxLength the request should be rejected (not trimmed) Business Facet(s) • unique = value must be 'unique' across the space defined in the field definition • systemId = System generated value; will not be set by API consumer
String36ForInteger64_Unique_SystemId	Number	Fundamental Facet(s) • Data passed as string to be persisted as integer (length cannot be 0) ○ MaxLength = 36 ○ Valid Characters = [0..9] • Persisted as Integer bits = 64 ○ Inferred 'Reject' = after the valid characters are checked if the value is larger than the MaxLength the request should be rejected (not trimmed) Business Facet(s)

		• unique = value must be 'unique' across the space defined in the field definition • systemId = System generated value; will not be set by API consumer
Token40_WhitespaceCollapse_Trim	String	Fundamental Facet(s) • MaxLength = 40 • Whitespace = contiguous sequences of #x20's are collapsed to a single #x20, and leading and trailing #x20's are removed • Trim = after the whitespace rules are applied if the value is larger than the MaxLength that data should be trimmed to the language relevant side of the data
Token40_WhitespaceCollapse_Unique	String	Fundamental Facet(s) • MaxLength = 40 • Whitespace = contiguous sequences of #x20's are collapsed to a single #x20, and leading and trailing #x20's are removed • Inferred 'Reject' = after the whitespace rules are applied if the value is larger than the MaxLength the request should be rejected (not trimmed) Business Facet(s) • unique = value must be 'unique' across the space defined in the field definition
Token64_WhitespaceCollapse_Unique	String	Fundamental Facet(s) • MaxLength = 64 • Whitespace = contiguous sequences of #x20's are collapsed to a single #x20, and leading and trailing #x20's are removed • Inferred 'Reject' = after the whitespace rules are applied if the value is larger than the MaxLength the request should be rejected (not trimmed) Business Facet(s) • unique = value must be 'unique' across the space defined in the field definition

References:

- *The basis for this model can be found in http://www.w3.org/TR/xmlschema-2/; where this is a base data type and some facets applied to the base data type. This representation is not meant to imply physical data model but rather the data validation rules associated with each field.*
- *Identity sizes derived from: http://en.wikipedia.org/wiki/Integer_(computer_science); they can be adjusted to a larger size up for any architectural optimizations.*

Note to Development:

- *Identity and Number sizes are represented as a method to provide the 'minimal expected requirements' from the business perspective; if there is technical reason such as ease of consumption or implementation to go beyond that type it should be mentioned and the value can be changed.*

API Enumeration Values

Domain Type - Values

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Account Access Level</td><td>Descriptor for the Access Level of the User Context

<table>
<tr><th>API Value</th><th>DB Value</th><th>Description</th></tr>
<tr><td>FULL</td><td>Full</td><td>User can do deposits to the account</td></tr>
<tr><td>VIEW</td><td>View</td><td>User can only view balance and history information on the account</td></tr>
<tr><td>(not present)</td><td>None</td><td>User is not provided information for this account
• This value will as currently defined will never be returned through the API, an Accounts are removed as part of the SQL.</td></tr>
</table>
</td></tr>
<tr><td>Account Holder Association Statuses</td><td>The list of Issuer Role Association Types

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>ACTIVE</td><td></td></tr>
<tr><td>INACTIVE</td><td></td></tr>
</table>
</td></tr>
<tr><td>Account Type</td><td>The account types are described by the following

<table>
<tr><th>API Value</th><th>DB Value</th><th>Description</th></tr>
<tr><td>POINT</td><td>PTS</td><td>The account type represents a point value</td></tr>
<tr><td>UNIT</td><td>UNIT</td><td>The account type represents a unit value</td></tr>
</table>
</td></tr>
<tr><td>Account Holder Features</td><td>(see section 'Feature Descriptions' below)</td></tr>
<tr><td>Beneficiary Features</td><td>(see section 'Feature Descriptions' below)</td></tr>
<tr><td>Credit Card Type</td><td>The credit card types are described by the following

<table>
<tr><th>API Value</th><th>DB Value</th><th>Description</th></tr>
<tr><td>AMEX</td><td>American Express</td><td></td></tr>
<tr><td>BCCARD</td><td>BC Card</td><td></td></tr>
<tr><td>ELECTRON</td><td>Electron</td><td></td></tr>
<tr><td>DINERSCLUB</td><td>Diners Club</td><td></td></tr>
<tr><td>DISCOVER</td><td>Discover</td><td></td></tr>
<tr><td>JCB</td><td>JCB</td><td></td></tr>
<tr><td>MAESTRO</td><td>Maestro</td><td></td></tr>
<tr><td>MASTERCARD</td><td>Mastercard</td><td></td></tr>
<tr><td>VISA</td><td>Visa</td><td></td></tr>
<tr><td>UNKNOWN</td><td>(Other)</td><td>If any other value is returned the API should return UNKNOWN</td></tr>
</table>
</td></tr>
<tr><td>Credit Debit Type</td><td>Descriptor for the Access Level of the User Context

<table>
<tr><th>API Value</th><th>Description</th></tr>
<tr><td>Cr</td><td>Credit</td></tr>
<tr><td>Dr</td><td>Debit</td></tr>
</table>
</td></tr>
<tr><td>Country Codes</td><td>The following is the subset of reference https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

(see ISO for data)</td></tr>
<tr><td>Currency Codes</td><td>The following is the subset of reference https://en.wikipedia.org/wiki/ISO_4217</td></tr>
</table>

	that is supported as Currency Codes available to the issuer. **Currency** / **Code** / **Number** / **Minor Unit** / **Symbol** Mexican Peso / MXN / 484 / 2 / $ US Dollar / USD / 840 / 2 / $ No Currency ➢ Currently used for UNIT / XXX / 999 / 0 /
Feature Statuses	(see section 'Feature Descriptions' below)
Image Statuses	The list of Issuer Role Association Types **Value** / **Description** ACTIVE INACTIVE
Issuer Supported Locale Codes	The following is the subset of reference http://www.oracle.com/technetwork/java/javase/javase7locales-334809.html that is supported as Language Tags available to the issuer. • language is the ISO-639 2 letter language code • territory is the ISO-3166 2 letter country code **Value** / **Description** en_US / English | United States es_MX / Spanish | Mexico ???? / ?????
Funding Result Types	The list of Funding Result Types **Value** / **Description** APPROVED / The Funding attempt was approved DECLINED-ISSUER / The Funding attempt was declined by rules or systems that enforce issuer rules. DECLINED-CREDITDEBITCARD / The Funding attempt was declined by rules or systems that enforce Credit/Debit card payment processor rules.
Payment Method Types	The list if user Payment Method Types **Value** / **Description** CREDITDEBITCARD
Payment Method Validity States	The list if user Payment Method Validity states **Value** / **Description** OK
Transaction Types	The following is a list of the transaction types. This list is a normalized list of transaction types as defined in ISO-8583 and the jPos specification. **Type** / **TypeCode** / **Description** 010000 / AUTHORIZATION 010031 / BALANCEINQUIRY 010032 / OPENTOBUYINQUIRY 020000 / PURCHASE 020002 / VOID 020020 / REFUND 020023 / RELOAD 022000 / COMPLETION
Time Zones	The following is the subset of reference https://en.wikipedia.org/wiki/List_of_tz_database_time_zones

<table>
<tr><td></td><td>(see ISO for data)</td></tr>
<tr><td>US State Codes</td><td>The following is the subset of reference https://en.wikipedia.org/wiki/ISO_3166-2:US

(see ISO for data)</td></tr>
<tr><td>User Association Mechanisms</td><td>The list of Issuer Relationship Statuses

<table><tr><th>Value</th><th>Description</th></tr><tr><td>ADMINISTRATION-EMAILADDRESS</td><td></td></tr><tr><td>BENEFACTOR-EMAILADDRESS</td><td></td></tr><tr><td>USER-DATAENTRY</td><td></td></tr></table></td></tr>
<tr><td>User Association Role</td><td>The list of Issuer Relationship Roles

<table><tr><th>Value</th><th>Description</th></tr><tr><td>ACCOUNTHOLDER</td><td>A relationship where the user and the accountholder represent the same person.</td></tr><tr><td>BENEFACTOR</td><td>A relationship where the accountholder has granted another person who is also a user of the systems the ability to access a subset of his/her account information.</td></tr></table></td></tr>
<tr><td>User Association Statuses</td><td>The list of Issuer Role Association Types

<table><tr><th>Value</th><th>Description</th></tr><tr><td>ACTIVE</td><td></td></tr><tr><td>INACTIVE</td><td></td></tr></table></td></tr>
</table>

accountholders | AH&BF

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource
GET	/accountholders/{accountHolderId}
GET	/beneficiarys/{accountHolderId}

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{accountHolderId}	Pattern_Numeric	id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
3. {if resources is /accountholders/{accountHolderId} only}
    a. Make private call to API userAssociations to get data required to do the following:
        i. Confirm signed on user has required '**userAssociationRoletype**=ACCOUNTHOLDER' with ***{accountHolderId}***; else error (according to error matrix) and exit noun resolution
        ii. Confirm user association with account holder has '**userAssocationRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
        iii. Confirm user association with account holder has '**accountHolderRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
4. OUT OF SCOPE - Make private call to API accountholder-features to get data required to do the following:
    a. Confirm the feature API Action and Request URL combination is a member of is '0=ENABLED'; else error (according to error matrix) and exit noun resolution

    ---
5. Noun data resolved to via URI parameter ***{accountHolderId}*** to query entry "686" in sample query

Relation data resolution

Sample query to achieve functional goal

```
SELECT
      cardholder.id AS id,
      cardholder.issuer AS issuer_id,
      issuer.name as issuer_name,
      issuer.localeId as issuer_localeId,
      cardholder.firstName AS givenName,
      cardholder.lastName AS familyName,
      cardholder.phone AS phoneNumber,
      cardholder.email AS emailAddress
FROM
      cardholder,
      issuer
WHERE
      issuer.id = cardholder.issuer and
      cardholder.id = 686
```

Query Parameters (URL Parameter Keys are case sensitive)

	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	cardholder.id	(A) - below
issuer	Yes-ReadOnly	object				
.id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	cardholder.issuer	(A) - below
.name	Yes-ReadOnly	string	Yes	Token40_WhitespaceCollapse_Unique • Uniqueness = System-wide	issuer.name	
.localeId	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Issuer Supported Locale Codes	issuer.localeId	
givenName	Yes-ReadOnly	string	No	Token40_WhitespaceCollapse_Trim	cardholder.firstName	
familyName	Yes-ReadOnly	string	Yes	Token40_WhitespaceCollapse_Trim	cardholder.firstLast	
phoneNumber	Yes-UpdateOnly	string	No	Regex20_PhoneNumber	cardholder.phone	
emailAddress	Yes-UpdateOnly	string	No	Regex255_EmailAddress	cardholder.email	

Comments

A. 'id' is intentionally treated as a JSON Data Type 'string' to allow for changing to a GUID without having to change the API

accounts | AH&BF

Related API UR

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource
GET	/accountholders/{accountHolderId}/accounts
GET	/accountholders/{accountHolderId}/accounts/{accountId}
GET	/beneficiarys/{accountHolderId}/accounts
GET	/beneficiarys/{accountHolderId}/accounts/{accountId}

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{accountHolderId}	Pattern_Numeric	(none)
{accountId}	Pattern_Numeric	id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
3. Make private call to API userAssociations to get data required to do the following:
    a. Confirm signed on user has required '**userAssociationRoletype**' for URL segment [ACCOUNTHOLDER, BENEFACTOR] with ***{accountHolderId}***; else error (according to error matrix) and exit noun resolution
    b. Confirm user association with account holder has '**userAssocationRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
    c. Confirm user association with account holder has '**accountHolderRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
4. OUT OF SCOPE - Make private call to API accountholder-features to get data required to do the following:
    a. Confirm the feature API Action and Request URL combination is a member of is '0=ENABLED'; else error (according to error matrix) and exit noun resolution

    ---
5. Noun data resolved to via URI parameter ***{accountHolderId}*** to query entry "686" in sample query
    a. URI parameter ***{accountId}*** to query entry "100000000051 " in sample query

Relation data resolution

Sample query to achieve functional goal

- ***'account.active'*** is equal to value for TRUE; this value is intentionally not part of the response
- ***'account.onlineAccess'*** is equal one of the DB equivalent values of the display values for [View Only, Full Access]

```
SELECT
    cardholder_accounts.account AS id,
    account.accountType AS accountType,
    plan.name AS name,
    account.endDate AS endDate,
    (SELECT
        SUM(CASE
            WHEN te.debit = 'Y' THEN te.amount * - 1
            WHEN te.debit = 'N' THEN te.amount
            ELSE 0
            END) AS col_0_0_
        FROM
            txn tx
                CROSS JOIN
            txn_entry te
        WHERE
            tx.id = te.txn
                AND te.account = account.id
                AND te.valid = 'Y'
                AND tx.retcode = '00'
        GROUP BY te.account) AS balanceAvailable,
    currency.currencyCodeAlpha3 AS currencyCodeAlpha3,
    currency.currencyCodeNumeric AS currencyCodeNumeric,
    plan.endDate AS rules_endDate,
    plan.onlineAccess AS rules_accesslevel
FROM
    cardholder_accounts,
    account,
    plan,
    currency
WHERE
    account.id = cardholder_accounts.account and
    account.active = 1 and  /* comment out to test */
    plan.id = account.plan and
    plan.onlineAccess in ('view' , 'full') and  /* comment out to test */
    currency.id = plan.currency and
    cardholder_accounts.cardholder = 686 /* change value '686' to test */
    and
    account.id = 100000000051 /* change value '100000000051 ' to test */
ORDER BY
    plan.name
```

Default data sort

- Noun Field: name (asc)
    - *Note this when displayOrderGroup becomes live data the sort will change to [displayOrderGroup (asc), name (asc)

Query Parameters (URL Parameter Keys are case sensitive)

- *These query parameters only apply to `/accountholders/{accountHolderId}/accounts`*

Field or (Reference)	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			
accountType	**accountType**	• enum_text_ci	
name	**name**	• contains_text_ci	

**Notes:*

- *All URL parameters keys can be used in conjunction with each other unless designated by (- or -)*
- *When query parameters are referenced in and ERROR response, the field should be referenced as "Query Parameter: {URL Parameter Key}"*

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	account.id	(A) - below (B) - below
accountType	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Account Type	account.accountType	(C) - below
name	Yes-ReadOnly	string	Yes	Token40_WhitespaceCollapse_Unique • Uniqueness = per Issuer	plan.name	
endDate	Yes-ReadOnly	string	Yes	Date8601	account.endDate	
balanceAvailable	Yes-ReadOnly	number	Calc	Currency • Validation based on [currencyCodeNumeric]	(see query above)	
currencyCodeAlpha3	Yes-ReadOnly	string	Yes	String3_Enumeration • Domain Type = Currency Codes - Code field	currency.currencyCodeAlpha3	
currencyCodeNumeric	Yes-ReadOnly	string	Yes	String3_Enumeration • Domain Type = Currency Codes - Number field	currency.currencyCodeNumeric	
rules	Yes-ReadOnly	object				
.endDate	Yes-ReadOnly	string	Yes	Date8601	plan.endDate	
.accessLevel	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Account Access Level	plan.onlineAccess	
.loadAmountMin	Yes-ReadOnly	number	Yes	{TO BE DETERMINED}	{RESEARCH REQUIRED}	(D) - below
.loadAmountMax	Yes-ReadOnly	number	Yes	{TO BE DETERMINED}	{RESEARCH REQUIRED}	(E) - below
displayOrderGroup	Yes-ReadOnly	number	Yes	{TO BE DETERMINED}	{RESEARCH REQUIRED}	(F) - below

Comments

A. 'id' is intentionally treated as a JSON Data Type 'string' to allow for changing to a GUID without having to change the API
B. Current SaaS Cardholder site displays this field as 'Account #'
C. It is acknowledged that the API domain values do not match what is currently persisted in the database, we want the API values to be more explicit.
D. rules.loadAmountMin will be hard-coded to '2' as a number.
E. rules.loadAmountMax will be hard-coded to '500' as a number.
F. displayOrderGroup will be hard-coded to '1' as a number.

cards | AH only

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource
GET	/accountholders/{accountHolderId}/cards
GET, PUT	/accountholders/{accountHolderId}/cards/{cardId}

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{accountHolderId}	Pattern_Numeric	(none)
{cardId}	Pattern_Numeric	id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
3. Make private call to API userAssociations to get data required to do the following:
    a. Confirm signed on user has required '**userAssociationRoletype=**ACCOUNTHOLDER' with ***{accountHolderId}***; else error (according to error matrix) and exit noun resolution
    b. Confirm user association with account holder has '**userAssocationRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
    c. Confirm user association with account holder has '**accountHolderRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
4. OUT OF SCOPE - Make private call to API accountholder-features to get data required to do the following:
    a. Confirm the feature API Action and Request URL combination is a member of is '0=ENABLED'; else error (according to error matrix) and exit noun resolution

    ---
5. Noun data resolved to via URI parameter ***{accountHolderId}*** to data ***'card.cardholder'*** to identify the list of records in the ***'cards'*** table that are associated to the user

Relation data resolution

Sample query to achieve functional goal

```
SELECT
    card.id AS id,
    card.createdDate AS createdDate,
    card.startDate AS startDate,
    card.issuedDate AS issueDate,
    card.endDate AS expirationDate,
    card.active AS active,
    card.number AS USEDFORCALC_number
FROM
    card
WHERE
    card.cardholder = 686 /* change value '686' to test */
ORDER BY
    card.createdDate desc;
```

Default data sort

- Noun Field: createdDate (desc)

Query Parameters (URL Parameter Keys are case sensitive)

	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForString16_Unique_SystemId • Uniqueness = system-wide	card.id	(A) - below
createdDate	Yes-ReadOnly	string	Yes	Date8601	card.createdDate	
startDate	Yes-ReadOnly	string	Yes	Date8601	card.startDate	
issueDate	Yes-ReadOnly	string	Yes	Date8601	card.issuedDate	
expirationDate	Yes-ReadOnly	string	Yes	Date8601	card.endDate	
cardNumberLast4	Yes-ReadOnly	string	Calc	Pattern4_NumericOrMask	{calculated} - see comment (C)	(B) - below (C) - below
active	Yes-UpdateOnly	boolean	Yes	Boolean	card.active	

Comments

A. 'id' is intentionally treated as a JSON Data Type string to allow for changing to a GUID without having to change the API
B. It is acknowledged that many times 'id' and 'cardNumber' will be the same but the documentation should outline which field should be used for what purpose giving us more built for change option

C. Field 'cardNumberLast4' is resolved by the following steps:
    a. Determine the value for logical '`fullCardNumber`' value by the following:
        i. If field `card.number` is populated with a value then that value becomes the '`fullCardNumber`'
        ii. else `card.id` is the '`fullCardNumber`'
    b. cardNumberLast4 field equals the right most 4 digits '`fullCardNumber`'
        i. if '`fullCardNumber`' is less than 4 digits long then prefix the value with the character 'X' until the resulting string is 4 characters in length

cards-cardnumber | AH only

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource
GET	/accountholders/{accountHolderId}/cards/{cardId}/cardnumber

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{accountHolderId}	Pattern_Numeric	(none)
{cardId}	Pattern_Numeric	id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
3. Make private call to API userAssociations to get data required to do the following:
   a. Confirm signed on user has required '**userAssociationRoletype=**ACCOUNTHOLDER' with ***{accountHolderId}***; else error (according to error matrix) and exit noun resolution
   b. Confirm user association with account holder has '**userAssocationRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
   c. Confirm user association with account holder has '**accountHolderRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
4. OUT OF SCOPE - Make private call to API accountholder-features to get data required to do the following:
   a. Confirm the feature API Action and Request URL combination is a member of is '0=ENABLED'; else error (according to error matrix) and exit noun resolution

   ---
5. Noun data resolved to via URI parameter ***{accountHolderId}*** to data ***'card.cardholder'*** to identify the list of records in the ***'cards'*** table that are associated to the user

Relation data resolution

Sample query to achieve functional goal

- *Note: this is a subset of what is described for cards, DEV choice on if you want this portion to be shared/same service etc…*

```
The important thing in how this is resolved is that the card.id belongs to the referenced cardholer

SELECT
    card.id AS id,
    card.number AS USEDFORCALC_number
FROM
    card
WHERE
    card.cardholder = 686 /* change value '686' to test */
    AND card.id = "6039509991133432"  /* change value to test */
```

Default data sort

- Noun Field: createdDate (desc)

Query Parameters (URL Parameter Keys are case sensitive)

	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
cardNumber	Yes-ReadOnly	string	Calc	Pattern16_Numeric	{calculated} - see comment (A)	(A) - below (B) - below

Comments

A. Field '**cardNumber**' is resolved by the following steps:
   a. If field `card.number` is populated with a value then that value becomes the '**cardNumber**'
   b. else `card.id` is the '**cardNumber**'
B. It is acknowledged that initially the '**id**' and '**cardNumber**' field will be the same; but at this time we want this to operate as described

features | AH only

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource
GET	/accountholders/{accountHolderId}/features
GET	/accountholders/{accountHolderId}/features/{accountHolderFeatureCode}

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{accountHolderId}	Pattern_Numeric	(none)
{accountHolderFeatureCode}	(none)	(none)

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
3. Make private call to API userAssociations to get data required to do the following:
   a. Confirm signed on user has required '**userAssociationRoletype=**ACCOUNTHOLDER' with ***{accountHolderId}***; else error (according to error matrix) and exit noun resolution
   b. Confirm user association with account holder has '**userAssocationRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
   c. Confirm user association with account holder has '**accountHolderRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution

   ---
4. Noun data resolved to via ???? - (OUT OF SCOPE - currently returns each known feature with a value of '0=ENABLED')

Relation data resolution

- sample query to achieve functional goal
  - {TBD}

Default data sort

- Noun Field: (????)

Query Parameters (URL Parameter Keys are case sensitive)

	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields

Field Name	CONTEXT: user	GET Sort	GET Filter	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
accountHolderFeatureCode	Yes-ReadOnly	No	No	string	Yes	String64_Enumeration • Domain Type = Account Holder Features	{DOES NOT EXIST}	
featureStatus	Yes-ReadOnly	No	No	number	Yes	Number_Enumeration_MinInc0MaxInc 999 • Domain Type = Account Holder Features	{DOES NOT EXIST}	(A) - below

Comments

A. Hard Code to 'ACTIVE'

fundings-creditdebitcard | AH&BF

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource
POST	/accountholders/{accountHolderId}/fundings/creditdebitcard
POST	/accountholders/{accountHolderId}/fundings/creditdebitcard/{creditDebitCardId}
POST	/beneficiarys/{accountHolderId}/fundings/creditdebitcard
POST	/beneficiarys/{accountHolderId}/fundings/creditdebitcard/{creditDebitCardId}

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{accountHolderId}	Pattern_Numeric	(none)
{creditDebitCardId}	Pattern_Numeric	creditcard.id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
3. Make private call to API userAssociations to get data required to do the following:
   a. Confirm signed on user has required '**userAssociationRoletype**' for URL segment [ACCOUNTHOLDER, BENEFACTOR] with ***{accountHolderId}***; else error (according to error matrix) and exit noun resolution
   b. Confirm user association with account holder has '**userAssocationRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
   c. Confirm user association with account holder has '**accountHolderRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
4. OUT OF SCOPE - Make private call to API accountholder-features to get data required to do the following:
   a. Confirm the feature API Action and Request URL combination is a member of is '0=ENABLED'; else error (according to error matr ix) and exit noun resolution

---

5. DVH API for internal "Account Load Orders"

Relation data resolution

- Not available (resolved through internal API)

Default data sort

- Not applicable

Query Parameters (URL Parameter Keys are case sensitive)

	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields (requests only)

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
amount	Yes-CreateOnly	number	Yes	Currency_MinExc0 • Validation based on [amountCurrencyCodeNumeric]	(internal API)	
amountCurrencyCodeNumeric	Yes-CreateOnly	string	Yes	String3_Enumeration • Domain Type = Currency Codes - Number field	(internal API)	
accountId	Yes-CreateOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	(internal API)	
paymentMethod		object				
.creditDebitCard		object				
.number	Yes-CreateOnly	string	Yes	Pattern16_CreditCardNumber	(internal API)	
.expireMonth	Yes-CreateOnly	number	Yes	Number_Enumeration_Month	(internal API)	
.expireYear	Yes-CreateOnly	number	Yes	Number_Enumeration_Year+20	(internal API)	
.securityCode	Yes-CreateOnly	string	Yes	Pattern4_SecurityCode	(internal API)	
.nameOnCard	Yes-CreateOnly	string	No	String40_Trim	(internal API)	
.billingAddress		object				
.line1	Yes-CreateOnly	string	Yes	String40_Trim_Nonempty	(internal API)	
.line2	Yes-CreateOnly	string	No	String40_Trim	(internal API)	
.city	Yes-CreateOnly	string	No	String40_Trim	(internal API)	
.countryCode	Yes-CreateOnly	string	No	String2_Enumeration • Domain Type = Country Codes	(internal API)	

.state	Yes-CreateOnly	string	No	String20_Trim • Domain Type = US State Codes	(internal API)	
.postalCode	Yes-CreateOnly	string	Yes	String10_Trim_Nonempty	(internal API)	

**Note: only the following fields are applicable to POST/creditdebitcard/{creditDebitCardId}*

- amount
- amountCurrencyCodeNumeric
- accountId

Cross Field Validation Rules (prior to calling internal API)

POST

- **amount** field errors
  - error.code 3501| will occur if there is more than the maximum allowed minor unit value associated with the currencyCode value in field **amountCurrencyCodeNumeric** - see Enumerations (Currency Codes)
- **state** field errors
  - error.code 3502| will occur on if the field ***countryCode = US***; If this is true then the state field must also be a member of the list in ***"Domain Type = US State Codes"***

Business Rules (prior to calling internal API)

POST

Condition	Result
referenceName already exists	*Error Matrix* information for HTTP 422 with result = 4001
paymentMethod.creditDebitCard.number longer than 16	*Error Matrix* information for HTTP 422 with result = 4005
paymentMethod.creditDebitCard.number shorter than 15	*Error Matrix* information for HTTP 422 with result = 4005
paymentMethod.creditDebitCard.number fails Luhn check	*Error Matrix* information for HTTP 422 with result = 4005
paymentMethod.creditDebitCard.expireMonth smaller than 1	*Error Matrix* information for HTTP 422 with result = 4006
paymentMethod.creditDebitCard.expireMonth larger than 12	*Error Matrix* information for HTTP 422 with result = 4006
paymentMethod.creditDebitCard.expireYear smaller than current year value	*Error Matrix* information for HTTP 422 with result = 4006
paymentMethod.creditDebitCard.expireYear larger than current year value + 20	*Error Matrix* information for HTTP 422 with result = 4006
paymentMethod.creditDebitCard.securityCode length less than 3	*Error Matrix* information for HTTP 422 with result = 4008
paymentMethod.creditDebitCard.securityCode length greater than 4	*Error Matrix* information for HTTP 422 with result = 4008
amount is less than associated noun value for ***accounts: rules.loadAmountMin***	*Error Matrix* information for HTTP 422 with result = 4004
amount is greater than associated noun value for ***accounts: rules.loadAmountMin***	*Error Matrix* information for HTTP 422 with result = 4004

accountId not associated to accountholderId; resolution similar to
`SELECT`
`  cardholder_accounts.account AS id`
`FROM`
`  cardholder_accounts,`
`  account`
`WHERE`
`  account.id = cardholder_accounts.account and`
`  cardholder_accounts.cardholder = 686 and /* value representing accountholderId */`
`  account.id = 100000000051 /* value representing accountId */`
	Error Matrix information for HTTP 422 with result = 4000

accountId associated noun value "***rules.accessLevel*** <> FULL"	*Error Matrix* information for HTTP 422 with result = 4002
accountId associated value for value for ***accounts: currencyCodeNumeric <> amountCurrencyCodeNumeric***	*Error Matrix* information for HTTP 422 with result = 4003

Internal API calling

Processing a Funding request requires calling an internal "Account Load Orders API" that takes responsibility for encrypting protecting the credit card information. When getting a response from the process should do the following:

Internal API call response	Results in ICP API: respond with
HTTP response 201	(process to generate response according to noun definition below
HTTP response 400 with error ➢ "parameter accountNumber is not valid"	*Error Matrix* information for HTTP 422 with error.code = 4005
HTTP response 400 with error ➢ "parameter expireMonth is not valid"	*Error Matrix* information for HTTP 422 with error.code = 4006
HTTP response 400 with error: ➢ (any not listed above)	*Error Matrix* information for HTTP 500 with error.code = 1021
HTTP response 500	*Error Matrix* information for HTTP 500 with error.code = 1022
Other (any other result or no result)	*Error Matrix* information for HTTP 500 with error.code = 1022

Noun Fields (response only)

Field Name	CONTEXT: user	JSON Data	Value Required	Data Validation Type	DB Field/resolution	Comment

		Type				
result	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Funding Result Types	(see Resolution below)	
authorizationCode	Yes-ReadOnly	string	No	String32_Trim	(see Resolution below)	
amount	Yes-ReadOnly	number	Yes	Currency_MinExc0 • Validation based on [amountCurrencyCodeNumeric]	(set equal to corresponding value in API request)	
amountCurrencyCodeNumeric	Yes-ReadOnly	string	Yes	String3_Enumeration • Domain Type = Currency Codes - Number field	(set equal to corresponding value in API request)	
accountId	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	(set equal to corresponding value in API request)	(A) - below
transactionId	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	(see Resolution below)	(A) - below
declineReasons		object				
.issuer		array				
..code	Yes-CreateOnly	string	Yes	TBD: (will update prior to API being public)	{DOES NOT EXIST}	
..message	Yes-CreateOnly	string	Yes	TBD: (will update prior to API being public)	{DOES NOT EXIST}	
..field	Yes-CreateOnly	string	No	TBD: (will update prior to API being public)	{DOES NOT EXIST}	
.creditDebitCard		array				
..code	Yes-CreateOnly	string	Yes	TBD: (will update prior to API being public)	(none)	
..message	Yes-CreateOnly	string	Yes	TBD: (will update prior to API being public)	(none)	
..field	Yes-CreateOnly	string	No	TBD: (will update prior to API being public)	(none)	

Comments

A. 'id' is intentionally treated as a JSON Data Type string to allow for changing to a GUID without having to change the API

Resolution (for HTTP 201 internal API responses)

An approved "Account Load Orders API v.1.0 ('ALO' below)" will generate the following response (listed here to help in resol ution):

```
{
 "success" : true,
 "approval" : “123456”,
 "transactionId" : “00000000-0000-0000-0000-00000000000”,
 "amount" : 250.00,
 "paymentMethodToken" : "6585"
 "accountTxnId" : 6115,
}
```

result

API Value	Condition
APPROVED	if 'ALO.success = true'
DECLINED-CREDITDEBITCARD	if 'ALO.success = false'

authorizationCode

API Value	Condition
'ALO.approval'	if 'ALO.approval' is present in the response
null	if 'ALO.approval' is not present

transactionId

API Value	Condition
'ALO.accountTxnId'	if 'ALO.accountTxnId' is present in the response
null	if 'ALO.accountTxnId' is not present

A declined "Account Load Orders API v.1.0 ('ALO' below)" will generate the following response (listed here to help in resolu tion); according to 1.0 of the specification document the codes below are possible in responses

- Detailed error responses resulting from internal API request will be done later.

Declined response should look like this where we are focusing on returning the

HTTP=201


```
{
  "funding": {
    . . . (body)
    "declineReasons": {
      "creditDebitCard": [
        {
          "code": "5200",
          "message": "Declined Credit/Debit card",
          "field": ""
        }
      ],
      "issuer": []
    }
  },
  "operation": {
    "result": "OK",
    "errors": [],
    "requestTimeStampUtc": "2015-09-30T13:14:15Z",
    "responseTimeStampUtc": "2015-09-30T13:14:20Z"
  }
}
```

images | AH&BF

Related API UR

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource
GET	/accountholders/{accountHolderId}/images
GET	/accountholders/{accountHolderId}/images/{imageId}
GET	/beneficiarys/{accountHolderId}/images
GET	/beneficiarys/{accountHolderId}/images/{imageId}

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{accountHolderId}	Pattern_Numeric	(none)
{imageId}	Pattern_Numeric	id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
3. Make private call to API userAssociations to get data required to do the following:
   a. Confirm signed on user has required '**userAssociationRoletype**' for URL segment [ACCOUNTHOLDER, BENEFACTOR] with ***{accountHolderId}***; else error (according to error matrix) and exit noun resolution
   b. Confirm user association with account holder has '**userAssocationRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
   c. Confirm user association with account holder has '**accountHolderRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
4. OUT OF SCOPE - Make private call to API accountholder-features to get data required to do the following:
   a. Confirm the feature API Action and Request URL combination is a member of is '0=ENABLED'; else error (according to error matrix) and exit noun resolution

   ---
5. {TBD}

Relation data resolution

- sample query to achieve functional goal
  - {TBD}

Default data sort

- Noun Field: createdDate (desc)

Query Parameters (URL Parameter Keys are case sensitive)

	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForString16_Unique_SystemId • Uniqueness = system-wide	{DOES NOT EXIST}	(A) - below (B) - below
status	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Image Statuses	{DOES NOT EXIST}	(C) - below
imageUrl	Yes-ReadOnly	string	Yes	{TO BE DETERMINED}	{DOES NOT EXIST}	(D) - below
profileUrl	Yes-ReadOnly	string	Yes	{TO BE DETERMINED}	{DOES NOT EXIST}	(D) - below
thumbnailUrl	Yes-ReadOnly	string	Yes	{TO BE DETERMINED}	{DOES NOT EXIST}	(D) - below

Comments

A. 'id' is intentionally treated as a JSON Data Type string to allow for changing to a GUID without having to change the API
B. Return hard-coded response of "999999", until data persistence is available
C. Return hard-coded response of "ACTIVE", until data persistence is available
D. Return hard-coded response of data according to mapping below
   - TEMPORARY IMAGE RESOLUTION
     - Base values

baseURL	http://res.cloudinary.com/denicbb/image/upload/
profileURL	/c_thumb,g_face:center,h_120,r_max,w_120,z_0/a_0
thumbnailUrl	/c_thumb,g_face:center,h_60,r_max,w_60,z_0/a_0

     - Image to CardholderID mapping

681	/shawnedwards_l6tjic

683	/deni_luwywn.jpg
682	/andyh_a8wnfb
686	/Taran_orig_ytljie.jpg
(any other)	{no image available}

- How to resolve each field in responswe
  - id = 999999 {hard-coded}
  - status = ACTIVE {hard-coded}
  - imageUrl = baseUrl+imageName(mapped to CardholderId)
  - profileUrl = baseUrl+profileUrl+imageName(mapped to CardholderId)
  - thumbnailUrl = baseUrl+thumbnailUrl+imageName(mapped to CardholderId)
- If the Id field does not exist in the image mapping above it should resolve in

GET /beneficiarys/{accountHolderId}/images	[(empty array)]
GET /beneficiarys/{accountHolderId}/images/{imageId}	error matrix row for error.code=1003

transactions | AH&BF

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource
GET	/accountholders/{accountHolderId}/transactions
GET	/accountholders/{accountHolderId}/accounts/{accountId}/transactions
GET	/beneficiarys/{accountHolderId}/transactions
GET	/beneficiarys/{accountHolderId}/accounts/{accountId}/transactions

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{accountHolderId}	Pattern_Numeric	(none)
{accountId}	Pattern_Numeric	transactionentry.account.id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
3. Make private call to API userAssociations to get data required to do the following:
   a. Confirm signed on user has required '**userAssociationRoletype**' for URL segment [ACCOUNTHOLDER, BENEFACTOR] with ***{accountHolderId}***; else error (according to error matrix) and exit noun resolution
   b. Confirm user association with account holder has '**userAssocationRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
   c. Confirm user association with account holder has '**accountHolderRelationshipStatus**=ACTIVE'; else error (according to error matrix) and exit noun resolution
4. OUT OF SCOPE - Make private call to API accountholder-features to get data required to do the following:
   a. Confirm the feature API Action and Request URL combination is a member of is '0=ENABLED'; else error (according to error matr ix) and exit noun resolution

   ---
5. Noun data resolved to via URI parameter ***{accountHolderId}*** to data ***'txn_entry.cardHolder*** to identify the list of records in the ***'txn_entry'*** table that are associated with
   a. The list of records that would fit the resolution so far is then restricted to only the records that meet the following crite ria:
      i. ***'txn.retCode'*** is equal to DB value for '00'
      ii. ***'txn_entry.valid'*** is equal to value for TRUE

Relation data resolution

Sample query to achieve functional goal

```
SELECT
      txn.id AS id,
      txn.date AS transactionDateTime,
      txn.type AS transactionType,
      txn.settled AS settled,
      txn.settledDate AS settleDateTime,
      merchant.id AS merchant_id,
      merchant.name AS merchant_name,
      card.id AS USEDFORCALC_cardId,
      card.number AS USEDFORCALC_cardNumber,
      txn_entry.debit AS transactionEntry_creditDebitType,
      currency.currencyCodeAlpha3 AS transactionEntry_currencyCodeAlpha3,
      currency.currencyCodeNumeric AS transactionEntry_currencyCodeNumeric,
      txn_entry.amount AS transactionEntry_amount,
      txn_entry.account AS transactionEntry_account_id,
      account.accountType AS transactionEntry_account_accountType,
      plan.name AS transactionEntry_account_name
FROM
      txn_entry
INNER JOIN txn ON txn.id = txn_entry.txn
LEFT JOIN merchant ON merchant.id = txn.merchant
LEFT JOIN card ON card.id = txn.card
INNER JOIN account ON account.id = txn_entry.account
INNER JOIN plan ON plan.id = account.plan
INNER JOIN currency ON currency.id = plan.currency
WHERE
      txn_entry.valid = 'Y' and
      txn.retCode = 00 and
      txn_entry.cardHolder = 686 /* change value '686' to test */
ORDER BY
      txn.date desc, txn.id desc
```

Default data sort

- Noun Field: transactionDateTime (desc), id (desc)

Query Parameters (URL Parameter Keys are case sensitive)

Field or (Reference)	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)	**page** - and - **pageSize**	• paging_page • paging_pageSize	
transactionDateTime	**transactionDateTime** **min_transactionDateTime**	• match_Date8601 • minRange_Date8601	

	- or, and - **max_transactionDateTime**	• maxRange_Date8601	
merchant.name	**merchantName**	• contains_text_ci	

Notes:

- *All URL parameters keys can be used in conjunction with each other unless designated by (- or -)*
- *When query parameters are referenced in and ERROR response, the field should be referenced as "Query Parameter: {URL Parameter Key}"*

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	txn.id	(A) - below
transactionDateTime	Yes-ReadOnly	string	Yes	DateTimeUtc	txn.date	
transactionType	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Transaction Types (Type column)	txn.type and {see - Type Calculation below}	
transactionTypeName	Yes-ReadOnly	string	Calc	Token64_WhitespaceCollapse_Unique • Uniqueness = System-wide and Domain Type = Transaction Types	{see - Type Calculation below}	
transactionTypeCode	Yes-ReadOnly	string	Calc	String32_Enumeration • Domain Type = Transaction Types (Type Code column)	{see - Type Calculation below}	
settled	Yes-ReadOnly	boolean	Yes	Boolean	txn.settled	
settleDateTime	Yes-ReadOnly	string	No	DateTimeUtc	txn.settledDate	
merchant	Yes-ReadOnly	object				
.id	Yes-ReadOnly	string	No	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	merchant.id	(A) - below
.name	Yes-ReadOnly	string	No	Token40_WhitespaceCollapse_Unique • Uniqueness = per Issuer	merchant.name	
cardNumberLast4	Yes-ReadOnly	string	Calc	Pattern4_NumericOrMask	{calculated} - see comment (D)	(C) - below
transactionEntry	Yes-ReadOnly	object				
.creditDebitType	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Credit Debit Type	txn_entry.debit	
.currencyCodeAlpha3	Yes-ReadOnly	string	Yes	String3_Enumeration • Domain Type = Currency Codes - Code field	currency.currencyCodeAlpha3	
.currencyCodeNumeric	Yes-ReadOnly	string	Yes	String3_Enumeration • Domain Type = Currency Codes - Number field	currency.currencyCodeNumeric	
.amount	Yes-ReadOnly	number	Yes	Currency • Validation based on [currencyCodeNumeric]	txn_entry.amount	(B) - below
.account	Yes-ReadOnly	object				
..id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	txn_entry.account	(A) - below
..accountType	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Account Type	account.accountType	
..name	Yes-ReadOnly	string	Yes	Token40_WhitespaceCollapse_Unique • Uniqueness = per Issuer	plan.name	

Comments

A. 'id' is intentionally treated as a JSON Data Type string to allow for changing to a GUID without having to change the API
B. The returned value should always be a positive about sign for the value should be determined based on Credit/Debit from the user's perspective for the application's purpose
C. Field 'cardNumberLast4' is resolved by the following steps:
    a. Determine the value for logical '`fullCardNumber`' value by the following:
        i. If field `card.number` is populated with a value then that value becomes the '`fullCardNumber`'
        ii. else `card.id` is the '`fullCardNumber`'
    b. cardNumberLast4 field equals the right most 4 digits '`fullCardNumber`'

i. if '`fullCardNumber`' is less than 4 digits long then prefix the value with the character 'X' until the resulting string is 4 characters in length

Type Calculation

- The API will then be responsible for using the field txn.type the this mapping table to generate the relevant fields in the API response when it is a defined standard type and just returning the text as is for the "CUSTOM" types.
    - The default/culture-neutral (en-US) transactionTypeName is listed in this table; the API will be responsible for returning this in its localized terminology based on the API Accept-Language header(s) unless it is a CUSTOM type then the database value will be returned

DB txn.type	API transactionType	API transactionTypeCode	API transactionTypeName
AUTH	AUTH	020000	Purchase
REPEATAUTH	REPEATAUTH	020000	Repeat Purchase
PURCHASE	PURCHASE	020000	Purchase
REFUND	REFUND	020020	Refund
RELOAD	RELOAD	020023	Reload
ADDVALUE	ADDVALUE	020021	Add Money
OTB	OTB	010032	Open To Buy
BALANCEINQUIRY	BALANCEINQUIRY	010031	Balance Inquiry
PLANRESET	PLANRESET	020022	Plan Reset
Initial_Credit	INITIALCREDIT	020029	Initial Credit
Initial_Debit	INITIALDEBIT	020019	Initial Debit
(any other value)	CUSTOM	(empty string)	(value in field txn.type) • *Note: no localization*

images

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource	Feature
GET, POST	/images	(none)
PUT, DELETE	/images/{imageId}	(none)

Note: POST and PUT actions are performed with content-type being sent as empty.

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
   ---
3. INWORK BELOW HERE STILL
   a. CONFIRM USER HAS SIGNED ON????
4. Noun data resolved to via URI parameter ***{accountHolderId}*** to data ***'card.cardholder'*** to identify the list of records in the ***'user'*** table that are associated to the user

Relation data resolution

Sample query to achieve functional goal

```
SELECT
    ??? AS id,
    ??? AS imageUrl
FROM
    ???
WHERE
    user_association.userId = 111
ORDER BY
    createdDate
```

Default data sort

- Noun Field: createdDate (asc)

Query Parameters (URL Parameter Keys are case sensitive)

These query parameters only apply to GET /userassociations

Field or (Reference)	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Different behavior from other APIs

POST & PUT

- **The content type will not be JSON and the API is expecting it to be empty/not set; any other value in the contentType should result in http response=415**
- The body will only contain an **image** field that is multi-part form data

Field Validation

POST & PUT

- **image** field errors

HTTP Response	error.code	Named error from error matrix (condition)
400	3008	Image field cannot be empty.
400	3009	Image field does not allow this file format.
400	3010	Image field size has exceeded maximum {maxSize}

  - Initial allowed file types were set to:
    - JPEG
    - JPG
    - PNG
  - Initial {maxSize} was set to: 50kb

Business Rules (prior to calling internal API)

POST

- **imageId** field errors
  - The {maxMembers} count for this collection = 1

http=422	error.code = 4007	POST failed maximum member count {maxMembers}

Noun Fields (responses only)

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForString16_Unique_SystemId	{DOES NOT EXIST}	(A) - below

				• Uniqueness = system-wide		
status	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Image Statuses	{DOES NOT EXIST}	(B) - below
imageUrl	Yes-ReadOnly	string	Yes	{TO BE DETERMINED}	{DOES NOT EXIST}	(D) - below
profileUrl	Yes-ReadOnly	string	Yes	{TO BE DETERMINED}	(see Resolution below)	(D) - below
thumbnailUrl	Yes-ReadOnly	string	Yes	{TO BE DETERMINED}	(see Resolution below)	(D) - below

Comments

A. 'id' is intentionally treated as a JSON Data Type string to allow for changing to a GUID without having to change the API
B. Return hard-coded response of "ACTIVE", until data persistence is available

Resolution

- **profileUrl** - will be imageUrl concatenated with cloudinary attributes /c_thumb,g_face:center,h_120,r_max,w_120
- **thumbnailUrl** - will be imageUrl concatenated with cloudinary attributes /c_thumb,g_face:center,h_600,r_max,w_60

issuers

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource	Feature
GET	/issuers	(none)
GET	/issuers/{issuerId}	(none)

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{issuerId}	Pattern_Numeric	id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL

   Note: this intentionally does not confirm user sign on

Relation data resolution

Sample query to achieve functional goal

```
SELECT
      issuer.id as issuerId,
      issuer.name as issuerName,
      issuer.localeId as issuerLocaleId,
      issuer.tz as timezone_id
FROM
      issuer

ORDER BY
      issuer.name
```

Default data sort

- Noun Field: issuer.Name (asc)

Query Parameters (URL Parameter Keys are case sensitive)

Field or (Reference)	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	issuer.id	(A) - below
name	Yes-ReadOnly	string	Yes	Token40_WhitespaceCollapse_Unique • Uniqueness = System-wide	issuer.name	
localeId	Yes-ReadOnly	string	Yes	String64_Enumeration • Domain Type = Issuer Supported Locale Codes	issuer.localeId	
timezone	Yes-ReadOnly	object				
.id	Yes-ReadOnly	string	Yes	String40_Enumeration • Domain Type = Time Zones	issuer.localeId	
.currentUtcOffset	Yes-ReadOnly	string	Calc	Pattern6_TimeZone	{calculated} - see comment (B)	

Comments

A. 'id' is intentionally treated as a JSON Data Type 'string' to allow for changing to a GUID without having to change the API
B. The API will be responsible for determining the current offset value where current means as of the time of the request

issuers-graphicalassets

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource	Feature
GET	/issuers/{issuerId}/graphicalassets	(none)

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{issuerId}	Pattern_Numeric	id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL

   Note: this intentionally does not confirm user sign on

Relation data resolution

Sample query to achieve functional goal

```
SELECT
      issuer.id as id,
      issuer.imageName as logoUrl,
      issuer.hexColorCode as accestHexColorCode
FROM
      issuer
WHERE issuer.id = 1
```

Default data sort

- Noun Field: issuer.Name (asc)

Query Parameters (URL Parameter Keys are case sensitive)

Field or (Reference)	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	issuer.id	(A) - below
logoUrl	Yes-ReadOnly	string	Yes	{TO BE DETERMINED}	issuer.imageName	
accentHexColorCode	Yes-ReadOnly	string	Yes	{TO BE DETERMINED}	issuer.hexColorCode	

Comments

A. 'id' is intentionally treated as a JSON Data Type 'string' to allow for changing to a GUID without having to change the API

paymentmethods

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource	Feature
GET	/paymentmethods	(none)

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
   ---
3. Confirm user has signed on
   a. limit result data to what is linked to user reference

Relation data resolution

Sample query to achieve functional goal

```
SELECT
   creditcard.id AS id,
   "CREDITDEBITCARD" AS paymentMethodType,
   creditcard.nickName AS referenceName,
   creditcard.lastFour AS referenceValue,
   UPPER(creditcard.type) AS creditDebitCard_typeName, /*note: this is really a mapping to types; not UPPER*/
   "OK" AS validState
FROM
   creditcard
INNER JOIN user_association ON user_association.cardHolder = creditcard.cardHolder
WHERE
   user_association.user = unhex('210525FC36DD44709958CE8B6CC97AA0') /*note: signed on user reference*/
ORDER BY
      creditcard.nickName ASC
```

Default data sort

- Noun Field: creditcard.nickname (asc)

Query Parameters (URL Parameter Keys are case sensitive)

These query parameters only apply to GET /userassociations

Field or (Reference)	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			
paymentMethodType	**paymentMethodType**	enum_text_ci	
validState	**validState**	enum_text_ci	

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	creditcard.id	(A) - below
paymentMethodType	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Payment Method Types	(resolved via source of data: table = creditcard)	(B) - below
referenceName	Yes-ReadOnly	string	Yes	Token64_WhitespaceCollapse_Unique • Uniqueness = userId	creditcard.nickName ➢ if Null or emptyString then creditDebitCard.typeName equivalent	
referenceValue	Yes-ReadOnly	string	Yes	String32_Trim	creditcard.lastFour	
validState	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Payment Method Validity States	{DOES NOT EXIST}	(C) - below

Comments

A. 'id' is intentionally treated as a JSON Data Type 'string' to allow for changing to a GUID without having to change the API
B. If the source of the data is the table = 'creditcard' then the value for this field is set to "CREDITDEBITCARD"
C. Return hard-coded response of "OK", until data persistence is available

paymentmethods-creditdebitcards

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource	Feature
GET, POST	/paymentmethods/creditdebitcards	(none)
GET, PUT, DELETE	/paymentmethods/creditdebitcards/{creditDebitCardId}	(none)

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
   ---
3. Confirm user has signed on
   a. limit result data to what is linked to user reference
   ---
4. DVH API for internal "Payment Methods"

Relation data resolution

Sample query to achieve functional goal

```
SELECT
    creditcard.id AS id,
    creditcard.nickName AS referenceName,
    creditcard.lastFour AS referenceValue,
    "OK" AS validState,
    "(calculated from typeName)" AS creditDebitCard_type,
    UPPER(creditcard.type) AS creditDebitCard_typeName, /*note: this is really a mapping to types; not UPPER*/
    creditcard.lastFour AS creditDebitCard_numberLast4,
    creditcard.address1 AS billingAddress_line1,
    creditcard.address2 AS billingAddress_line2,
    creditcard.city AS billingAddress_city,
    creditcard.country AS billingAddress_countryCode,
    creditcard.state AS billingAddress_state_US,
    creditcard.province AS billingAddress_state_other,
    creditcard.zip AS billingAddress_postalCode
FROM
   creditcard
INNER JOIN user_association ON user_association.cardHolder = creditcard.cardHolder
WHERE
   user_association.user = unhex('210525FC36DD44709958CE8B6CC97AA0') /*note: signed on user reference*/
ORDER BY
      creditcard.nickName ASC
```

Default data sort

- Noun Field: creditcard.nickname (asc)

Query Parameters (URL Parameter Keys are case sensitive)

These query parameters only apply to GET /userassociations

Field or (Reference)	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields - request body

**Internal AP referenced is: POST …/cs.api/paymentmethods/creditdebitcards*

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	GET: creditcard.id	(A) - below
referenceName	Yes-CreateAndUpdate	string	Yes	Token40_WhitespaceCollapse_Unique • Uniqueness = userId	GET: creditcard.nickname ➢ if Null or emptyString then creditDebitCard.typeName equivalent POST: ➢ uniqueness validation: ➢ *internal API - field: referenceName	
referenceValue	Yes-ReadOnly	string	Calc	String32_Trim	creditcard.lastFour	
validState	Yes-ReadOnly	string	Yes	String32_Enumeration • Domain Type = Payment Method Validity States	{DOES NOT EXIST}	(D) - below
creditDebitCard		object				
.type	Yes-ReadOnly	string	Calc	String20_Enumeration • Domain Type = Credit Card Type	GET: creditcard.type	

.typeName	Yes-ReadOnly	string	Calc	String32_Trim	GET: {calculated} - see comment (B)	(B) - below
.numberLast4	Yes-ReadOnly	string	Calc	String32_Trim	GET: creditcard.lastFour	
.number	Yes-CreateOnly	string	Yes	Pattern16_CreditCardNumber	POST: *internal API - field: accountNumber	
.expireMonth	Yes-CreateOnly	number	Yes	Number_Enumeration_Month	POST: *internal API - field: expireMonth	
.expireYear	Yes-CreateOnly	number	Yes	Number_Enumeration_Year+20	POST: *internal API - field: expireYear	
.nameOnCard	Yes-CreateOnly	string	No	String40_Trim	POST: *internal API - field: nameOnCard	
billingAddress		object				
.line1	Yes-CreateAndUpdate	string	Yes	String40_Trim_Nonempty	GET, PUT: creditcard.address1 POST: *internal API - field: address1	
.line2	Yes-CreateAndUpdate	string	No	String40_Trim	GET, PUT: creditcard.address2 POST: *internal API - field: address2	
.city	Yes-CreateAndUpdate	string	No	String40_Trim	GET, PUT: creditcard.city POST: *internal API - field: city	
.countryCode	Yes-CreateAndUpdate	string	No	String2_Enumeration • Domain Type = Country Codes	GET, PUT: creditcard.country POST: *internal API - field: countryCode	
.state	Yes-CreateAndUpdate	string	No	String20_Trim • Domain Type = US State Codes	GET, PUT: creditcard.state & creditcard.province - or - GET, PUT: creditcard.province POST: *internal API - field: State	(C) - below
.postalCode	Yes-CreateAndUpdate	string	Yes	String10_Trim_Nonempty	GET, PUT: creditcard.country POST: *internal API - field: postalCode	

Comments

A. 'id' is intentionally treated as a JSON Data Type 'string' to allow for changing to a GUID without having to change the API
B. The field value for "creditDebitCard.type" is used along with the Accept-Language data to determine the value
C. If 'billingAddress.countryCode' = US then the field 'creditcard.state' is applicable
    a. else field 'creditcard.province' is applicable
D. Return hard-coded response of "OK", until data persistence is available

Cross Field Validation Rules

POST & PUT

- **state** field errors
    - error.code 3502| will occur on if the field ***countryCode = US***; If this is true then the state field must also be a member of the list in ***"Domain Type = US State Codes"***

Business Rules - prior to calling internal API

Condition	Result
referenceName already existss	*Error Matrix* information for HTTP 422 with result = 4001
creditDebitCard.number longer than 16	*Error Matrix* information for HTTP 422 with result = 4005
creditDebitCard.number shorter than 15	*Error Matrix* information for HTTP 422 with result = 4005
creditDebitCard.number fails Luhn check	*Error Matrix* information for HTTP 422 with result = 4005
creditDebitCard.expireMonth smaller than 1	*Error Matrix* information for HTTP 422 with result = 4006
creditDebitCard.expireMonth larger than 12	*Error Matrix* information for HTTP 422 with result = 4006
creditDebitCard.expireYear smaller than current year value	*Error Matrix* information for HTTP 422 with result = 4006
creditDebitCard.expireYear larger than current year value + 20	*Error Matrix* information for HTTP 422 with result = 4006

Business Rules - after calling internal API

Processing a Funding request requires calling an internal "Account Load Orders API" that takes responsibility for encrypting protecting the credit card information. When getting a response from the process should do the following:

Internal API call response	Results in ICP API: respond with
HTTP response 201	*Error Matrix* information for HTTP 201 with result = OK
HTTP response 400 with error ➢ "parameter accountNumber is not valid"	*Error Matrix* information for HTTP 422 with error.code = 4005
HTTP response 400 with error	*Error Matrix* information for HTTP 422 with error.code = 4006

➢ "parameter expireMonth is not valid"	
HTTP response 400 with error: ➢ (any not listed above)	*Error Matrix* information for HTTP 500 with error.code = 1019
HTTP response 500	*Error Matrix* information for HTTP 500 with error.code = 1020
Other (any other result or no result)	*Error Matrix* information for HTTP 500 with error.code = 1020

user

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource	Feature
GET, PUT	/user	(none)

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
   ---
3. Confirm user has signed on
   a. limit result data to what is linked to user reference

Relation data resolution

Sample query to achieve functional goal

```
SELECT
   user.email as emailAddress,
   user.givenName as givenNane,
   user.familyName as familyName,
   user.localeId as localeId
FROM
   user
WHERE
   user.id = unhex('210525FC36DD44709958CE8B6CC97AA0') /*note: signed on user reference*/
```

Default data sort

- Noun Field: creditcard.nickname (asc)

Query Parameters (URL Parameter Keys are case sensitive)

These query parameters only apply to GET /userassociations

Field or (Reference)	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
emailAddress	Yes-ReadOnly	string	Yes	{TO BE DETERMINED}	user.email	
givenName	Yes-UpdateOnly	string	No	String40_Trim	user.givenName	
familyName	Yes-UpdateOnly	string	No	String40_Trim	user.familyName	
localeId	Yes-UpdateOnly	string	No	String64_Enumeration • Domain Type = Issuer Supported Locale Codes	user.localeId	

Comments

userassociations

Related API URI

See http://docs.tcpv1userapi.apiary.io/#reference for request/response detail

Action	Resource	Feature
GET, POST	/userassociations	(none)
GET, PATCH, DELETE	/userassociations/{userAssociationId}	(none)

URL Parameter data validation

Parameter	Data Validation Type	Noun field resolution
{userAssociationId}	Pattern_Numeric	id

Noun Resolution

1. OUT OF SCOPE - Application authentication and resolution
2. OUT OF SCOPE - Confirm Application has authorization to the URL
   ---
3. Confirm user has signed on
   a. limit result data to what is linked to user reference

Relation data resolution

Sample query to achieve functional goal

```
SELECT
      user_association.id as id,
      user_association.role as userAssociationRole,
      user_association.mechanism as userAssociationMechanism,
      user_association.userStatus as userAssociationStatus,
      user_association.accountHolderStatus as accountHolderAssociationStatus,
      user_association.cardHolder as accountHolderId,
      cardholder.firstname as accountHolderGivenName,
      cardholder.lastname as accountHolderFamilyName,
      issuer.id as issuerId,
      issuer.name as issuerName,
      issuer.localeId as issuerLocaleId
FROM
      user_association
INNER JOIN cardholder ON cardholder.id = user_association.cardHolder
INNER JOIN issuer ON issuer.id = cardholder.issuer
WHERE
      user_association.user = unhex('210525FC36DD44709958CE8B6CC97AA0')
ORDER BY
      issuer.name,
user_association.role
```

Default data sort

- Noun Field: issuer.Name (asc), user_assocation.role (asc)

Query Parameters (URL Parameter Keys are case sensitive)

These query parameters only apply to GET /userassociations

Field or (Reference)	URL Parameter key(s)	Key query type & validation	Order By Parameters supported
(Paging Method)			
userAssociationRole	**userAssociationRole**	enum_text_ci	
userAssociationStatus	**userAssociationStatus**	enum_text_ci	
accountHolderAssociationStatus	**accountHolderAssociationStatus**	enum_text_ci	
issuer.id	**issuerId**	equals_id_string	

Noun Fields

Field Name	CONTEXT: user	JSON Data Type	Value Required	Data Validation Type	DB Field/resolution	Comment
id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	user_association.id	(A) - below
userAssociationRole	Yes-CreateOnly	string	Yes	String32_Enumeration • Domain Type = User Association Role	user_association.role	
userAssocationMechanism	Yes-CreateOnly	string	Yes	String64_Enumeration • Domain Type = User Association Mechanisms	user_association.mechanism	
userAssociationStatus	Yes-CreateAndUpdate	string	Yes	String32_Enumeration • Domain Type = User Association Statuses	user_association.status	
accountHolderAssociationStatus	Yes-CreateAndUpdate	string	Yes	String32_Enumeration • Domain Type = Account Holder Association Statuses	user_association.status	
accountHolder	Yes-ReadOnly	object				

.id	Yes-CreateOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	cardholder.id	(A) - below
.givenName	Yes-ReadOnly	string	Yes	Token40_WhitespaceCollapse_Trim	cardholder.firstName	
.familyName	Yes-ReadOnly	string	Yes	Token40_WhitespaceCollapse_Trim	cardholder.firstLast	
issuer	Yes-ReadOnly	object				
.id	Yes-ReadOnly	string	Yes	String36ForInteger64_Unique_SystemId • Uniqueness = System-wide	cardholder.issuer	(A) - below
.name	Yes-ReadOnly	string	Yes	Token40_WhitespaceCollapse_Unique • Uniqueness = System-wide	issuer.name	
.localeId	Yes-ReadOnly	string	Yes	String64_Enumeration • Domain Type = Issuer Supported Locale Codes	issuer.localeId	

Comments

A. 'id' is intentionally treated as a JSON Data Type 'string' to allow for changing to a GUID without having to change the API

Business Rules

- POST
  - accountholder.Id must already exist
  - In the initial implementation we want to enforce the following combinations are unique
    - accountholder.Id
    - userAssociationRole=ACCOUNTHOLDER

Private APIs - used by other context APIs

There will be a private API call within this noun that allows other userContext APIs to request data.

- Expectation for this call is that it would only return information related to this noun; not make decisions for the calling API.
- Expectation for this call could be done in a way that allows any of the following optional filtering parameters:
  - userAssociationRole
  - userAssociationStatus
  - issuerId
  - accountholder.id
- Current expected behavior
  - return the noun (or subset of the noun) based on any filtering provided in order for calling API to make decisions; as of cur rent behavior minimally this data will be needed
    - userAssociationRole
    - userAssociationStatus
    - issuer.id
    - accountholder.id