MAKING ELECTRONIC TRANSACTIONS **SECURE**

HST Issuer Server

API Guide

MISC-HST-ISSUER-SERVER-SPEC

Version 3.8.8 (25.07)

Summary

1. **Introduction** 4

1.1. **Backward Compatibility** 5

2. **Connectivity** 5

2.1. **URL Scheme** 6

2.2. **Key Management** 6

2.3. **Software Architecture and Technology** 6

3. **Onboarding HST Environment** 6

3.1. **SIK components** 8

4. **Application Program Interfaces (APIs) - Outbound** 9

4.1. **CheckEligibility** 9

4.2. **DigitizationNotification** 20

4.3. **SendPassCode** 28

4.4. **LifeCycleNotification** 34

4.5. **DeviceBindingEligibility** 38

4.6. **ChangeCardInfoNotification** 43

4.7. **GetSystemStatus** 47

4.8. **PushProvisioningNotification** 47

4.9. **BulkProvisionNotification** 50

5. **Tokenization BUS - Inbound** 52

5.1. **GetAssociatedTokens** 52

5.2. **GetTokenInfo** 58

5.3. **ChangeTokenStatus** 63

5.4. **ActivateToken** 67

5.5. **GetPANByPANRefID** 70

5.6. **ChangeCardInfo** 72

5.7. **GetSystemStatus** 76

5.8. **PushCard** 77

5.9. **BulkProvision** 81

5.9.1. **BulkProvision – Initiate Bulk Job** 81

5.9.2. **BulkProvision – Get Bulk Job Status** 82

6. **General Objects** 84

6.1. **CardMetaData** 84

6.2. **TokenInfo** 86

6.3. DeviceInfo....89
6.4. AuthenticationMethod....89
6.5. ExpirationDate....91
6.6. CardInfo....91
6.7. EncryptedPayload....92
6.8. TokenUserInfo....93
6.9. MerchantInfo....93
6.10. RiskInformation....93
6.11. TermsAndConditions....94
6.12. Market....95
6.13. CardProfile....95
6.14. BillingAddress....96
6.15. Provider....96
6.16. ClientInformation....98
6.17. PushNotification....100
6.18. ReturnObject....101
6.19. ErrorObject....101
6.20. ChipData....102
6.21. ChipDataValidationResult....102
7. Return Codes....103
8. Revision History....105

1. Introduction

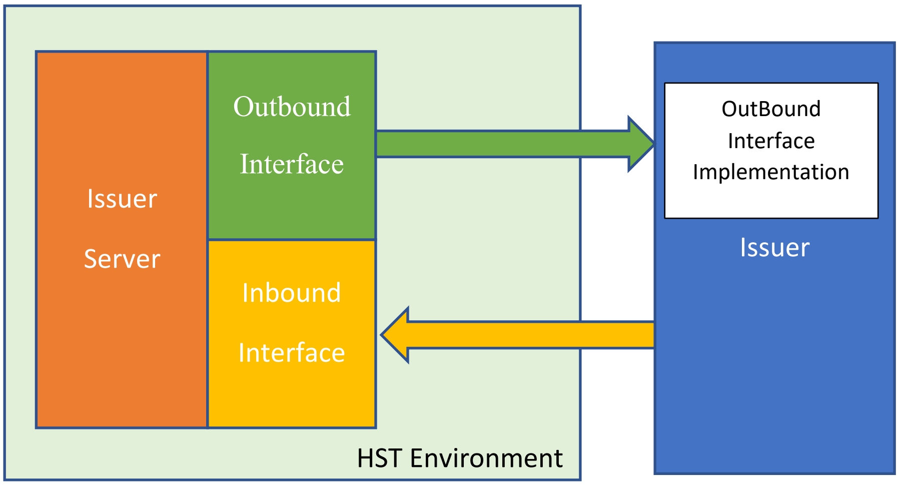
Issuer Server is the system responsible for:

- Authenticating the cardholder during card digitization;
- Defining card metadata during digitization;
- Managing digitized card lifecycle;
- Receiving notifications about digitized cards;
- Providing reports and statistics about the tokenization system;
- Providing detailed information about cards and tokens for support and troubleshooting purposes.

The Issuer Server connects to Visa VTS, Mastercard MDES, American Express and PL Vaults on behalf of the issuers. It also provides an Inbound and an Outbound interface to each Issuer connected to the ecosystem.

The Inbound interface allows Issuers to send life cycle commands to manage digitized cards and to inquiry the system about cards and tokens.

The Outbound interface is used to define cardholder authentication during card digitization and to notify Issuers about token status changes.



Outbound interface implemented by Issuer

1.1. Backward Compatibility

HST ensures that, whenever it is possible, changes to APIs are backward-compatible. The purpose of backward compatibility is to ensure that an API change is seamless and will not impact its utilization in the Issuer environment, in the same way the brands (Visa, Mastercard and American Express) promote updates on their environments for such existent's APIs, guaranteeing the minimal impact possible.

The following changes are considered as backward compatible:

- Adding a new API;
- Adding a new optional request or response element parameter to an existing API;
- Adding a new Enum value;
- Changing the order in which parameters are returned in existing APIs responses.

And for the scenarios above, the Issuer must continue accepting requests and not consider error when a new field is included.

2. Connectivity

The inbound and outbound APIs are designed as RPC style stateless webservices where each API endpoint represents an operation service published that only can be performed using **JSON** payload format. All strings in request and response are UTF-8 encoded and may have a version number API, which allows multiple versions of concurrent APIs to be deployed simultaneously.

Table 01 defines the supported HTTP response codes.

Error Code	Description
200	Success
400	Invalid request
401	Request Denied
403	Not allowed
404	Not found
500	Internal server error
501	Not implemented
503	Service not available

Table 1 – HTTP Response code

2.1. URL Scheme

The URL API follows the scheme bellow:
scheme://host[:port]/version/apiName

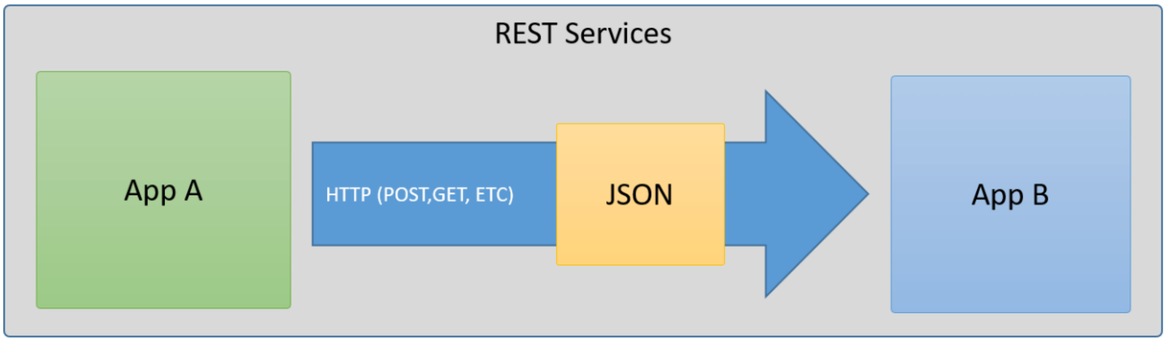
URL ELEMENT	Definition
Scheme	HTTPS
host[:port]	Described in the sections below
Version	v3
API	

2.2. Key Management

The process of exchanging client/server certificates for the establishment of mutual authentication in TLS 1.2 will be performed by HST (Compliance) and Issuers during the project initial steps.
There is a specific procedure to follow to initiate the certificates exchange process that will be shared with the responsible contact of the Issuer. All the communication will be performed using the kms@hst.com.br e-mail.

2.3. Software Architecture and Technology

The inbound and outbound APIs must be implemented/invoked using **REST API JSON** style.



Implementation using SOAP (XML schemas) **MUST NOT** be used.

3. Onboarding HST Environment

Definition of Parameters

- **Financial Institution Code:** Unique Code defined by HST during Issuer Onboarding that identifies the Issuer at HST Pay Token Services and is out of the scope of this document.
- **Sensitive Information Key (SIK):** It is an AES key generated by HST in its HSM during onboarding and shared between Issuer and HST through kms@hst.com.br e-mail explaining the process.

Notes:

1. Information about the SIK used in testing environment and other dynamic parameters for each issuer will be provided in a specific document.
2. The EncryptedData used in the JSON examples provided in this document were ciphered using the following testing SIKs:

- **AES-128 key type:** “404142434445464748494A4B4C4D4E4F”;
- **AES-256 key type:** “404142434445464748494A4B4C4D4E4F4F4E4D4C4B4A4948474645444342414 0”.

3.1. SIK components

For **AES-256** the issuer could combine three components by logical XOR operation.

Component 1:
E0E3A481C2E3D1E88E93773B6961B25FCE3A32E23BB0A042075DE2E9E9F15D61
Key Check Value (KCV): CF842B
XOR
Component 2:
B9E81FBDA791DE3AD18AB72F1CE2FB5F4B3C558777659B35BA5F32A49BE86FCC
KCV: ED0AE3
XOR
Component 3:
194AF97F2137499517508A5F39CE074FCA482A29079F723FFA449509315B73ED
KCV: AED09F

SIK (AES 256):
404142434445464748494A4B4C4D4E4F4F4E4D4C4B4A49484746454443424140
KCV (AES): 05E63C

For **AES-128** the issuer could combine three components by logical XOR operation.

Component 1: E0E3A481C2E3D1E88E93773B6961B25F
KCV: A2114B
XOR
Component 2: B9E81FBDA791DE3AD18AB72F1CE2FB5F
KCV: DCC7E1
XOR
Component 3: 194AF97F2137499517508A5F39CE074F
KCV: 900959

SIK (AES 128): 404142434445464748494A4B4C4D4E4F
KCV: 189956

4. Application Program Interfaces (APIs) - Outbound

The Outbound interface functions are called during card digitization, when an Issuer has to be notified about a token status change or to authenticate a user and retrieve available cards associated to the user.

4.1. CheckEligibility

This API is used by Issuer Server to inquiry the Issuer if a card is eligible for digitization. During this process, the card data (Cardholder Name, PAN, CVV and Expiration Date) must be validated by the Issuer. The real PAN must be associated with the TokenRefID or PANRefID elements, because in future calls the actual PAN may not be received. Issuer can **deny** digitization, **approve** it, or approve it with the requirement of additional cardholder identity validation (**ID&V**). In the case Issuer requests ID&V, it must return one or more ID&V methods available for the cardholder.

During the digitization process, there are two final provisioning status that indicates the initial condition of the token when cardholder first tried to provision:

- Yellow Flow: tokens that are initially issued in an “*inactive*” status and are stepped up for ID&V. The issuer must return the value **“85” – card is eligible for digitization and cardholder must be verified** on the *returnCode* element to present to the cardholder the ID&V methods available for identity validation. Cardholder will receive one or more options (*Call Center, App to App and OTP*) depending on Issuer implementation to choose after the card digitization. Cardholder must follow the process till the token activation.
- Green flow: tokens issued in an “*active*” status and no ID&V is performed. Normally applied when the Issuer already has authenticated the cardholder. The issuer must return the value **“00” – card is Eligible for digitization** on the *returnCode* element, and the card is activated right after digitization. For Issuer Wallets who requests cardholder authentication during enrollment or for Merchants which require COF or E-COM Tokens.

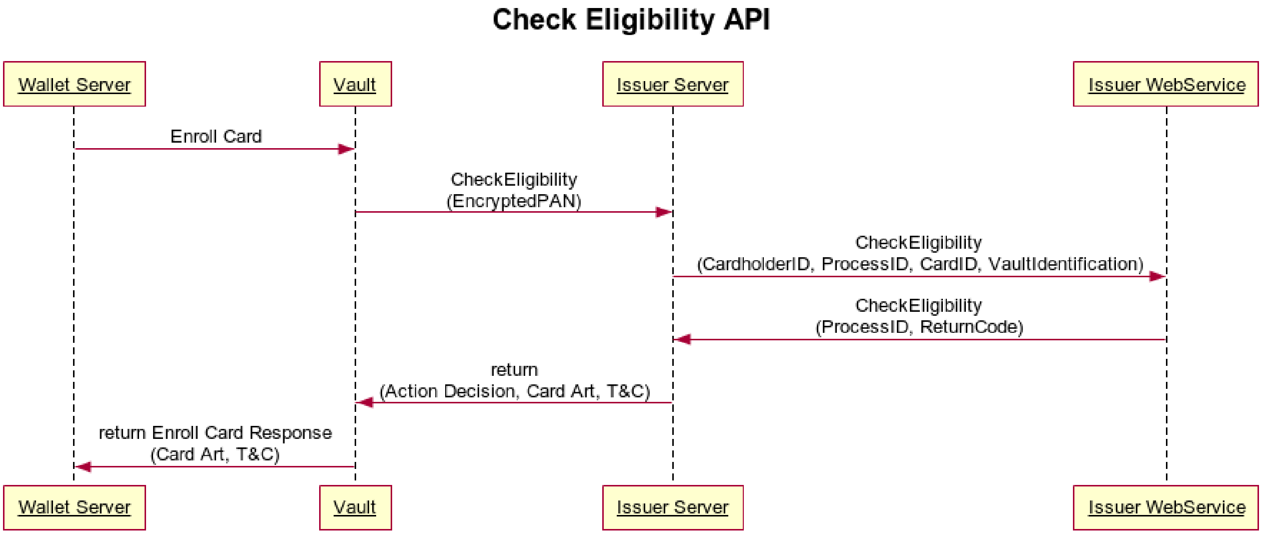
The Check Eligibility API also enables the issuer to associate the card being digitized to an internal cardholder identification, typically a bank account or a preexisting user identification. The identification gives more flexibility to issuers on future calls to identify customers and cards associated to them on HST Environment.

It is recommended to be one of the first APIs to be implemented during a I-TSP/TR-TSP project for a provisioning flow.

Additionally, some Wallet Providers can send, as Token Requestors, cardholder Risk Data to support the Issuers in the decision making for card digitization (green/yellow flow or denial). This information can optionally be provided in the *riskInformation* element.

Depending on the brand, HST can invoke a second (subsequent) call of Check Eligibility API to be able to provide this information to the Issuer.

The expected time for response of this API is approximately 2.0 seconds during the requests after being called. Otherwise, the Vault may receive a timeout and the provisioning will be failed.


Check Eligibility API
Wallet Server
Vault
Issuer Server
Issuer WebService
Enroll Card
CheckEligibility (EncryptedPAN)
CheckEligibility (CardholderID, ProcessID, CardID, VaultIdentification)
CheckEligibility (ProcessID, ReturnCode)
return (Action Decision, Card Art, T&C)
return Enroll Card Response (Card Art, T&C)
Wallet Server
Vault
Issuer Server
Issuer WebService

API endpoint	Method
Sandbox: https://{sandbox-issuer-host:port}/api/v3/checkeligibility	POST
Production: https://{issuer-host:port}/api/v3/checkeligibility	POST

CheckEligibilityRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Size:	64
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes

Element:	**vaultIdentification**

Description:	Possible values are:
	“VTS” – for Visa;
	“MDES” – for Mastercard;
	“AMEX” – for Amex;
	“PL” – for Private Label.
	Used to identify the Vault in case tokenRefID does not exist in HST database.

Type:	String
Size:	32
Required:	Yes
Element:	**walletID**
Description:	Identifier of the wallet that generated the request. Some wallet providers such Apple Pay and Google pay define it with the same value used for device identification. For Apple Pay, this field shows the SEID.
Type:	String
Size:	64
Required:	Optional
Example:	Format at HST WhiteLabel Wallet: N3GN-KWH6-NTYC-QNKN
Element:	**tokenRequestorID**
Description:	Identification of the Token Requestor requesting digitization. It identifies SamsungPay, ApplePay, a Multi Issuer Wallet or an Issuer Wallet. All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project.
Type:	String
Size:	64
Required:	Yes
Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data, it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PANRefID for further use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others. *Note: Please be aware that VTS will provision the same tokenRefID for a HCE token when a new digitization of the same card occurs within seven days from the token deletion by the cardholder.*

Type:	String
Size:	64
Required:	Optional Present for “VTS” and “MDES” Not present for “AMEX”
Element:	**PANRefID**
Description:	The PANRefID is a value assigned by the vault to identify the PAN. It identifies the PAN on the Vault. For VISA, each PAN generates a PANRefID value, which means a VISA PAN must have only one PANRefID value assigned. For Mastercard, the PANRefID it is associated to the Token Requestor, which means it is not unique for a PAN and it can have multiples PANRefIDs. By using this data, it is not necessary to input the real PAN value. It is recommended to relate the PANRefID value with the TokenRefID for further use in APIs such as GetAssociatedTokens, GetPANByPANRefID and others.
Type:	String
Size:	64
Required:	Present for “VTS” Optional for “MDES” Not present for “AMEX”
Element:	**encryptedCardInfo**
Description:	Encrypted CardInfo. Contains of card information to be used on digitization process.
Type:	EncryptedPayload
Required:	Yes
Element:	**processID**
Description:	Digitalization process identifier generated for each request. For Mastercard, this field must contain the same value during a complete digitization process, and it is sent on the next APIs such as DigitizationNotification and SendPassCode. For Visa and Amex, this value does not remain the same.
Type:	String
Size:	Max 64
Required:	Yes
Element:	**userLanguage**
Description:	User preferred language according to ISO 639 Version 3 Language Code (for example: “eng”).
Type:	String
Size:	3
Required:	Present for “VTS” Optional for “MDES” Not present for “AMEX”

Element: **Source**

Description: How the card number was obtained.
Possible values are:
“ON_FILE” – PAN origin is a card number stored in a merchant;
“MANUALLY” – PAN was entered by the customer;
“MOBILE_APP” – PAN provided by a mobile app. Typically a list of cards provided by the issuer after cardholder authentication;
“TOKEN” – The source of pan of this token (ECOM o COF) provisioning was issued by a token device bound (NFC/SE). Applicable to a scenario such as a wallet has a NFC/SE token and it is provisioning a new E-Commerce/COF token.
“BROWSER” – Indicates that the account details were pulled from a browser for tokenization.
“CONTACTLESS_TAP” - PAN was captured using "Tap to Add Card" service

Type: String

Required: Yes

Element: **riskInformation**

Description: Risk data provided by the Wallet Provider or by the Vault. This information can help the Issuer in the decision for card eligibility (green/yellow flow or denial).

Type: RiskInformation Object

Required: Optional

Element: **riskInformationResubmission**

Description: Depending on the brand implementation, Token Requestor risk information can be received after issuer answers to CheckEligibility. In this case, HST invoques the CheckEligibility API a second time in order to present issuer with this information. If TRUE, this field indicates the call is a resubmission on the API. Depending on Issuer evaluation of the risk data, a different return code can be replied on the second call. On most cases Issuers will switch from a Green flow to a Yellow or Red flow. In case of Yellow Flow, authentication information should be provided on response.
The absence of this field means it’s the first call to the API.

Type: Boolean

Required: Optional

Element: **tokenType**

Description: Information provided by HST using the parameters sent by the Vault to inform the Issuer the token type requesting the digitization.
Possible values are:
“HCE”, “SE”, “COF”, “ECOM”, “QRCODE”.

Type: String

Required: Yes

Element: **tokenRequestorName**

Description: Identification of the Token Requestor name requesting digitization. It identifies a Multi Issuer Wallet, an Issuer Wallet or a Merchant, like Uber, Netflix, Adyen, Apple Pay, Samsung Pay and others.

Type: String

Required: Optional

Element: **recommendedDecision**

Description: A suggestion provided by HST to support Issuer during the decision flow. This value uses a combination of other values received by the Vault to create a decision suggested.
Issuer can use this value to determine a flow to the cardholders.
Possible values are:
“GREEN”, “YELLOW” or “RED”.

Type: String

Required: Yes

Element: **recommendedDecisionReasonCode**

Description: The code that explains the *recommendedDecision*. Possible values are:

Restricted to RED flow recommendation:
“0001” – Error due to the card digitized in too many devices.
“0002” – Too many consecutive incorrect attempts of digitization (Invalid CVV2 or Expiration Date).
“0003” – Token Requestor recommendation.
“0005” – Token provisioned abroad. (*)
“0006” – Account with suspicious transactions history
“0007” – Too many consecutive attempts of token device digitization
“0008” – Too many consecutive token device digitization
“0009” – Cardholder name mismatches with devices owner.
“0010” – Device score is too low
“0011” – Account score is too low
“0012” – High risk digitization detected

Not restricted to the RED flow:
“**0004**” - CVV2 is present for issuer validation.

(*) By default, this code recommends RED flow in *recommendedDecision* element. However, it could be mapped to RecommendedDecision=YELLOW, by opening a support ticket on HST.

Type: String

Required: Optional

Element: **recommendedDecisionReasonCodeList**

Description: An array with all the codes reported by the vault and/or by the wallet provider to support the token provision decisioning. The

	possible values are the same of the listed in the *recommendedDecisionReasonCode*.
Type:	String[32]
Required:	Optional
Element:	**chipData**
Description:	Encrypted ChipData Object. This parameter is present only when Issuer supports Tap to Add Card and issuer has decided to receive and validate chip data by themselves.
Type:	EncryptedPayload
Required:	Conditional. MDES only, when *Tap to Add Card* feature is enabled.
Element:	**chipDataValidationResult**
Description:	ChipDataValidationResult object, present only when Issuer supports Tap to Add Card and issuer has decided to use Mastercard On-behalf service (OBS) for chip data validation.
Type:	ChipDataValidationResult Object
Required:	Conditional. MDES only, when *Tap to Add Card* feature is enabled.
Element:	**deviceType**
Description:	"UNKNOWN", "MOBILE_PHONE", "TABLET", "WATCH", "PC".
Type:	String
Size:	Max 32
Required:	Optional

CheckEligibilityResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request returned by the Issuer.
Type:	String
Required:	Yes
Element:	**processID**
Description:	Digitalization process identifier generated for each request returned by the Issuer.
Type:	String
Size	Max 64
Required:	Yes
Element:	**returnCode**
Description:	Possible values are: **"00"** – Card is eligible for digitization; **"05"** – Card is not eligible for digitization; **"85"** – Card is eligible for digitization and cardholder must be verified; **"16"** – Card not found, invalid PAN; **"22"** – Invalid card security code; **"23"** – Invalid card Expiration date;

“**27”** – Too many attempts, suspected fraud. Return expected when element "recommendedDecisionReasonCode" value received in request is "0002".

For Amex, the following specific digitization denial reasons may be returned by the Issuer: “24”, “25” and “26”.
“24” - Card has not been activated, replaced, or renewed card has not been activated;
“25” - Non-whitelisted accounts when a market is at beta test phase;
“**26”** - Ineligible instant account/instant membership account provisioning.

Type: String
Required: Yes

Element: **errorDescription**
Description: Error description returned only in error conditions for troubleshooting purpose.
Type: String
Required: Optional

~~Element:~~ ~~**encryptedCardMetaData**~~
~~Description:~~ ~~Encrypted CardMetaData. In case the Issuer does not send this value during request, Issuer Server will send it as null to the Vault and the brand will use the data configurated on their Card Metadata Management tool.~~
~~Type:~~ ~~EncryptedPayload~~
~~Required:~~ ~~Optional for VTS and MDES~~
Deprecated – New implementations must use CardMetaData field

Element: **cardMetaData**
Description: CardMetaData. This element is not encrypted. In case the Issuer does not send this field, the token requestor will receive the information configured by the issuer in the vault platform (VCMM or Mastercard Connect).
For Mastercard, the only field permitted is “**productID**” (HST Parameter), that must match the “**issuerConfigId”** (Mastercard Parameter) defined on Mastercard Connect.
For Visa, it can be defined by “**productID**”, that must match the “**profileID**” (VISA Parameter) defined on the Visa Card Metadata Management Tool (VCMM) or by sending the cardArtID and termsAndConditionsID.
For Amex, it is required to return **productID**, **productName** and **productType** elements.
Type: CardMetaData
Required: Optional for “VTS” and “MDES”
Required for “AMEX”

Element:	**authenticationMethods**
Description:	Authentication methods list for specific card, if authentication needed. The possible values for implementation are OTP, Call Center and App-to-App and their details are described on the AuthenticationMethod description element.
Type:	Array <AuthenticationMethod>
Required:	Optional
Element:	**userID**
Description:	Issuer identification on the cardholder. Typically, an account or online banking user ID. Only for auditing purpose on HST’s system, there is no participation during the provisioning and transaction flows.
Type:	String
Required:	Optional
Element:	**market**
Description:	Market object. Indicates the market/region where the card was issued.
Type:	Market object
Required:	Required only for AMEX.
Element:	**expirationDate**
Description:	Card expiry date.
Type:	ExpirationDate object
Required:	Required only for AMEX.
Element:	**PANSequence**
Description:	Funding account PAN sequence. Examples: 00 (Default Value), 01, 02, 03.
Type:	String
Size:	2
Required:	Required only for AMEX.
Element:	**chipDataValidationResponse**
Description:	The Issuer validated chipData validation result code. Must present if issuer is validating chip data for *Tap to Add Card* feature). Must be one of the following values: “VALID” = Chip data provided is valid, “INVALID” = Chip data provided is not valid, “NOT_PROCESSED” = Unable to process Chip data validation (issuer temporarily unavailable).
Type:	String
Size:	32
Required:	Conditional, only for MDES.

JSON Examples

CheckEligibilityRequest

```
{
  "requestID": "2",
  "institutionCode": "HST",
  "vaultIdentification": "VTS",
  "walletID": "N3GN-KWH6-NTYC-QNKN",
  "tokenRequestorID": "42301999123",
  "tokenRefID": "DNITHE381502386342002358",
  "PANRefID": "V-3815023863409817870482",
  "encryptedCardInfo": {
    "algorithm": "aes-ccm128",
    "nonce": "a96b3e84232d573c6592ceda",
    "encryptedData":
"KV1Mgkv40Nt4yggF1Ka7osdIkyMSsVe8K3o9wpQpMRTGeiXV2I65fIYgjZY1IGEpj/A7+KX3XB8C4
Foo8tEZ5xxQXa2PRudQ9B9s9WZbWoANcyaDAdw7ix7CQUN4x2ps9+oe8UaLtwjKrbKEDFkCML9rE9O
oco7vMr7y+uAlZ2NazPoWwx5fcQkn",
    "MACLength": 16
  },
  "processID": "1643ef957-622d-4137-abdf-fa605e81e72c",
  "source": "MANUALLY",
  "riskInformation": {
    "recommendedDecision": "YELLOW",
    "deviceScore": "2",
    "accountScore": "2"
  },
  "tokenType": "NFC",
  "tokenRequestorName": "HSTPayWallet",
  "recommendedDecision": "YELLOW",
}
```

Where:

//Plain CardInfo Object Data:

```
{
  "PAN": "1111110000000003",
  "expirationDate": {
    "month": "11",
    "year": "2024"
  },
  "CVV2": "500",
  "cardholderName": "FRANCISCO PEREIRA"
}
```

CheckEligibilityResponse

```
{
  "requestID": "2",
  "processID": "1643ef957-622d-4137-abdf-fa605e81e72c",
  "returnCode": "85",
  "cardMetaData": [
    {
      "productID": "14454"
    },
    "authenticationMethods": [
      {
        "identifier": "125485644",
        "type": "bank_app",
        "maskedInfo": "Mobile Banking App",
        "sourceAddress": "com.DemoBank.DemoApp",
        "platform": "ANDROID"
      },
      {
        "identifier": "125485633",
        "type": "cell_phone",
        "maskedInfo": "XXX-XXX-1234"
      }
    ],
    "userID": "12345678909"
  }
```

Card Meta Data Implementation Options

Option 1 (Default):

- The field **cardMetaData** is not sent in the response of this API. The Vault will get the metadata information default loaded on their platform. It is most applicable for scenarios when the Issuer has one card art image for BIN.

Option 2 (for Mastercard and Visa):

- Define a value for the **productID** field for each card product. It is most applicable for scenarios when the Issuer has more than one card art image for BIN, most likely for account range.

For Mastercard (*maximum size: 10*)

```
"cardMetaData": {
  "productID": "9835210843"
}
```

For Visa (*maximum size: 32*):

```
"cardMetaData": {
  "productID": "246380983124"
}
```

Option 3 (for Visa only):

- The Issuer can define the metadata during the digitization by sending the color values and other information.

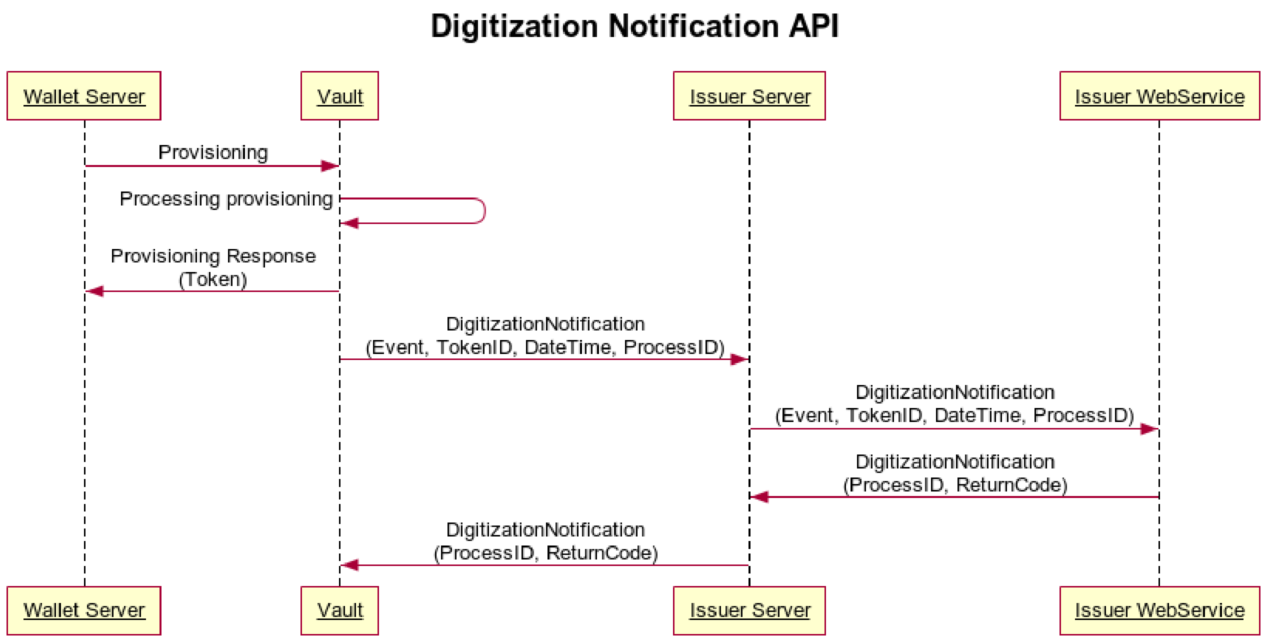
NOTE: All the parameters highlighted are optional for this Option, the Issuer can send only the cardArtId and termsAndConditionsId.

```
"CardMetaData": {
  "foregroundColor": "rgb(12,225,585)",
  "backgroundColor": "rgb(13,456,787)",
  "labelColor": "rgb(15,678,679)",
  "shortDescription": "Platinum",
  "longDescription": "Brand X Platinum Elite",
  "contactPhone": "98819838",
  "contactName": "Francisco Pereira",
  "cardArtId": "013",
  "termsAndConditionsId": "032"
}
```

4.2. DigitizationNotification

This API is used by Issuer server to send notifications to Issuer regarding the digitization process, therefore at the end of the process this API will be triggered **to inform the Issuer** the result of token creation process.

Note: For Mastercard, this is the only API that provides the complete token number associated to the card that was digitized. The calls to GetTokenInfo Inbound API will only retrieve the last 4 digits of the token for this card brand.



API endpoint	Method
Sandbox: https://{sandbox-issuer-host:port}/api/v3/digitizationnotification	POST
Production: https://{issuer-host:port}/api/v3/digitizationnotification	POST

DigitizationNotificationRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes

Element:	**processID**
Description:	Digitalization process identifier generated for each request. For Mastercard, this field must contain the same value during a complete digitization process, first generated on the CheckEligibility API, and it is sent on the next APIs, such as SendPassCode. For Visa and Amex, this value does not remain the same.
Type:	String
Size	Max 64
Required:	Yes
Element:	**vaultIdentification**

Description:	Possible values are:
	“VTS” – for Visa;
	“MDES” – for Mastercard;
	“AMEX” - for Amex;
	“PL” – for Private Label.
	Used to identify the Vault in case tokenRefID does not exist in HST database.

Type:	String
Required:	Yes
Element:	**walletID**
Description:	Identifier of the wallet that generated the request. Some wallet providers such Apple Pay and Google pay define it with the same value used for device identification. For Apple Pay, this field shows the SEID.
Type:	String
Size:	64
Required:	Optional
Example:	Format at HST WhiteLabel Wallet: N3GN-KWH6-NTYC-QNKN
Element:	**tokenType**
Description:	Possible values are: “HCE”, “SE”, ”COF”, ”ECOM”, ”QRCODE” (Case-Sensitive).
Type:	String
Required:	Required for “VTS” and “MDES” Not present for “AMEX”
Element:	**dateTime**
Description:	Format: yyyy-MM-ddTHH:mm:ss.SSS The value is required to be in GMT.
Type:	String
Required:	Yes
Element:	**Event**
Description:	Possible values: “CREATED”, “STAND_IN” (Case-Sensitive).
Type:	String
Required:	Yes
Element:	**tokenRequestorID**
Description:	

	Identification of the Token Requestor requesting digitization. It identifies SamsungPay, ApplePay, a Multi Issuer Wallet or an Issuer Wallet. All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project.
Type:	String
Size:	64
Required:	Yes
Element:	**tokenRequestorName**
Description:	Identification of the Token Requestor name requesting digitization. It identifies a Multi Issuer Wallet, an Issuer Wallet or a Merchant, like Uber, Netflix, Adyen, Apple Pay, Samsung Pay and others.
Type:	String
Size:	Max 64
Required:	Optional
Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data, it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PANRefID for further use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others.
Type:	String
Size:	64
Required:	Yes
Element:	**PANRefID**
Description:	The PANRefID is a value assigned by the vault to identify the PAN. It identifies the PAN on the Vault. For VISA, each PAN generates a PANRefID value, which means a VISA PAN must have only one PANRefID value assigned. For Mastercard, the PANRefID it is associated to the Token Requestor, which means it is not unique for a PAN and it can have multiples PANRefIDs. By using this data, it is not necessary to input the real PAN value. It is recommended to relate the PANRefID value with the TokenRefID for further use in APIs such as GetAssociatedTokens, GetPANByPANRefID and others.
Type:	String
Size:	64
Required:	Required for “VTS” and “MDES” Not present for “AMEX”

Element:	**encryptedCardInfo**
Description:	Encrypted CardInfo related to the card being digitized.
Type:	EncryptedPayload
Required:	Yes
Element:	**Source**
Description:	How the card number was obtained. Check “*CheckEligibility*” API for more details. Possible values are: **“ON_FILE”** – PAN origin is a card number stored in a merchant; **“MANUALLY”** – PAN was entered by the customer; **“MOBILE_APP”** – PAN provided by a mobile app. Typically a list of cards provided by the issuer after cardholder authentication; **“TOKEN”** – The source of pan of this token (ECOM o COF) provisioning was issued by a token device bound (NFC/SE). Applicable to a scenario such as a wallet has a NFC/SE token and it is provisioning a new E-Commerce/COF token. **“CONTACTLESS_TAP”** - PAN was captured using "Tap to Add Card" service.
Type:	String
Required:	Required for “VTS” and “MDES” Not present for “AMEX”
Element:	**actionResult**
Description:	Result of the digitization process. Possible values are: **“APPROVED”** – card was successfully tokenized; **“APPROVED_IDV”** – card was successfully tokenized and will need cardholder authentication for activation; **“INVALID_PAN”** – the card was not digitized due to the invalid PAN; **“INVALID_EXPIRATION_DATE”** – the card was not digitized due to the invalid expiration date; **“ISSUER_SYSTEM_ERROR”** – error on the issuer internal system; **“GENERIC_DECLINE”** – generic decline on the tokenization process; **“ERROR”** – error on the tokenization process.
Type:	String
Required:	Yes
Element:	**standInReasonCode**
Description:	Responsible to inform to the Issuer the reason why the digitization was entered in Stand-In flow in case the Issuer system did not respond. Possible values are: **“9020”** – Issuer system time outs; **“9027”** – CVV2 validate failure following VRM rules defined by the Issuer (ECIP RTD Decline); **“9216”** – Ineligible data for Token Type. Token is not a device based one;

	“9217” – Loyalty personalized data input is incorrect; **“9061”** – Switch detected error.
Type:	String
Required:	Optional for “VTS” and “MDES” Not present for “AMEX”
Element:	**termsAndConditions**
Description:	Information about the terms and conditions of the card.
Type:	TermsAndConditions
Required:	Optional
Element:	**token**
Description:	Encrypted TokenInfo of to the token created related to the card being digitized. For the MDES scenario, the Issuer will only receive the token information on this API.
Type:	EncryptedPayload
Required:	Optional

DigitizationNotificationResponse

Element:	**requestID**
Description:	Request identifier uniquely generated for each request returned by the Issuer.
Type:	String
Required:	Yes
Element:	**processID**
Description:	Digitalization process identifier generated for each request returned by the Issuer.
Type:	String
Size	Max 64
Required:	Yes
Element:	**returnCode**
Description:	Return Code: “00” for OK.
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

JSON Examples

DigitizationNotificationRequest

```
{
  "requestID": "4",
  "institutionCode": "HST",
  "processID": "1643ef957-622d-4137-abdf-fa605e81e72c",
  "vaultIdentification": "VTS",
  "walletID": "N3GN-KWH6-NTYC-QNKN",
  "tokenType": "HCE",
  "dateTime": "2015-05-18T14:40:32.000Z",
  "event": "CREATED",
  "tokenRequestorID": "42301999123",
  "tokenRefID": "DNITHE381502386342002358",
  "PANRefID": "V-3815023863409817870482",
  "encryptedCardInfo": {
    "algorithm": "aes-gcm256",
    "iv": "99aa57b5eb8dc1a8d0f91f40",
    "encryptedData":
"ACBh0D9ZD0k7v1M31uzTk/+7zSNEH9wML7cLi4reKjWcVXm1PFHTz9hxb0RIQdWYBoH7rzyNCHh9l
ZA//7O8BQRgpAIOTY5kgRINWqNiL0DlwKJ+obxGcwssFsBR45ByeiFFFTAk+gPlzM4h4Aj/oqdu4fp
+r0CHiZBTv19PmH4W12BA29lQXI+N",
    "MACLength": 16
  },
  "source": "MOBILE_APP",
  "actionResult": "APPROVED_IDV",
  "token": {
    "algorithm": "aes-ccm128",
    "nonce": "b3c0f84e500e50ffcd5f563e",
    "encryptedData":
"Q6sfnucc1f6duTMvzcUa5SueAKUeDpd2Fq+fcSg/xBFU0LhSoiTMJ/3BiZc6uP5GrWbUouoSr01ve
r9YiauDloy9hD4buW2ZiE24sguOpjhlsx2DyNX0ryBlJOjyhK/9z9dfQaRSwK6TxBmndsMAOCGRf5g
QiwiFdgF7w/xcJfoDrSnQ9MPkLThyIAA7+y+8ZLiFjjRJGAY1fXjoNnVjsDsxPuIq+p5hI0BrQ9YWH
CqCllbDX5PycBMT7e5jL2dgz4p7hP2fNrlmXY5EVqhPD12FbjSliXKNib4RdJe/xbol5WCzwhsxncu
+8Owt0VMzdZs6DdcrDcMMmB4l+5UAsrzx73JhkAhO0j5NK2u+llrwrAcn8Ul+A/tFv1W3HrarixA1X
PLVpGdOq+3DgjxqkLBZOV1WiZ0D+q0vtVrmkqUvvlyzZafcLufMw9/7KX1sONmvQDP+2zC1R96VghQ
Njj3wIo7xH/+T0TKhUMqwCapvxkSwD70l87z/eYPKmIb4YXWgbiyKnRUyhCnE5vDxYAlOt8+5mz0LY
nJtLAPEMvtyxmIsFU6GW+AYvVJb3ae9ZNfcdsK9DkHpEmHIQ0UffvEAv7ELgjZALWOV1AsxlHiBLJd
YxGXO+3BPuUJssFc1P99AXWyKOTY51KBJMVsWxHc=",
    "MACLength": 16
  }
}
```

Where:

//Plain CardInfo Object Data:

```
{
  "PAN": "1111110000000003",
  "expirationDate": {
    "month": "11",
    "year": "2024"
  },
  "cardholderName": "FRANCISCO PEREIRA"
}
```

And:

//Plain TokenInfo Object Data:

```
[
  {
    "token": "1111113245678979",
    "expirationDate": {
      "month": "10",
      "year": "2024"
    },
    "state": "ACTIVE",
    "type": "HCE",
    "lastTokenStatusUpdatedTimeStamp": "2015-05-18T14:40:32.000Z",
    "entityOfLastAction": "ISSUER",
    "deviceInfo": {
      "deviceType": "MOBILE_PHONE",
      "deviceNumber": "1234",
      "deviceName": "AndroidCellPhone",
      "serialNumber": "874759678487"
    },
    "OTPCodeIndicate": "PRESENT",
    "OTPCodeExpiration": "2015-05-18T14:40:32.000Z",
    "PANsLastFour": "1234",
    "previousPANsLastFour": "4653",
    "tokenRefID": "DNITHE381502386342002358"
  }
]
```

DigitizationNotificationResponse

```
{
  "requestID": "4",
  "processID": "1643ef957-622d-4137-abdf-fa605e81e72c",
  "returnCode": "00"
}
```

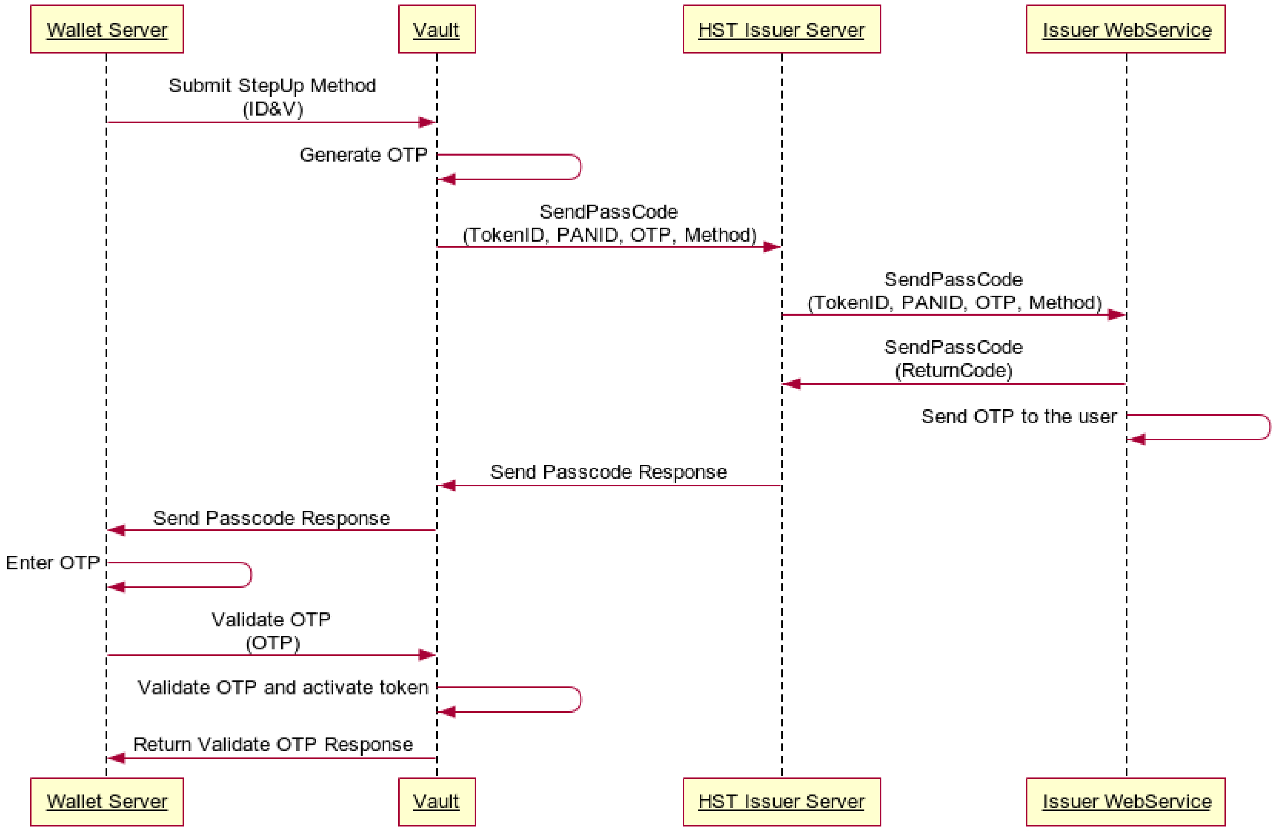
4.3. SendPassCode

This method is used when Issuer when issuer answers with return code “85” (Requires ID&V) REQUIRE_IDV on CheckEligibility or DeviceBindingEligibility and the cardholder selects “otp_sms” or “otp_email” as step-up methods. In this case the vault generates an OTP and requests the issuer to deliver the OTP to the related phone or email address.

The expected time for response of this API is approximately 2.0 seconds during the requests after being called, otherwise the Vault will receive timeout and the cardholder will get a failed message.

Note: Using the PANRefID or TokenRefID element as a parameter, the Issuer is able to identify the real card PAN and the respective cardholder that must receive the passcode.

Send Pass Code API



API endpoint	Method
Sandbox: https://{sandbox-issuer-host:port}/api/v3/sendpasscode	POST

Production: https://{issuer-host:port}/api/v3/sendpasscode	POST	

SendPassCodeRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes
Element:	**processID**
Description:	Digitalization process identifier generated for each request. For Mastercard, this field must contain the same value during a complete digitization process, first generated on the CheckEligibility API. For Visa and Amex, this value will not remain the same.
Type:	String
Size	Max 64
Required:	Yes
Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data, it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PANRefID for further use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others.
Type:	String
Size:	64
Required:	Required for “VTS” and “MDES” Not present for “AMEX”
Element:	**PANRefID**
Description:	The PANRefID is a value assigned by the vault to identify the PAN. It identifies the PAN on the Vault. For VISA, each PAN generates a PANRefID value, which means a VISA PAN must have only one PANRefID value assigned. For Mastercard, the PANRefID it is associated to the Token Requestor, which means it is not unique for a PAN and it can have multiples PANRefIDs.

Type:	By using this data, it is not necessary to input the real PAN value. It is recommended to relate the PANRefID value with the TokenRefID for further use in APIs such as GetAssociatedTokens, GetPANByPANRefID and others.
Size:	String
Required:	64
	Required for “VTS”
	Optional for “MDES”
	Not present for “AMEX”
Element:	**authenticationMethod**
Description:	Possible values are: “cell_phone”, “email”.
Type:	String
Required:	Yes
Element:	**OTP**
Description:	Authentication code.
Type:	String
Size:	16
Required:	Yes
Element:	**OTPExpiration**
Description:	Authentication code expiration time. Format: yyyy-MM-dd HH:mm:ss The value will be in GMT.
Type:	String
Required:	Yes
Element:	**vaultIdentification**

Description:	Possible values are:
	“VTS” – for Visa;
	“MDES” – for Mastercard;
	“AMEX” - for Amex;
	“PL” – for Private Label.
	Used to identify the Vault in case tokenRefID does not exist in HST database.

Type:	String
Required:	Yes
Element:	**walletID**
Description:	Identifier of the wallet that generated the request. Some wallet providers such Apple Pay and Google pay define it with the same value used for device identification. For Apple Pay, this field shows the SEID.
Type:	String
Size:	64
Required:	Optional
Example:	Format at HST WhiteLabel Wallet: N3GN-KWH6-NTYC-QNKN

Element:	**otpReasonCode**
Description:	The possible values are: **"PAYMENT"** **"CARDHOLDER_STEPUP"** **"DEVICE_BINDING"**
Type:	String
Required:	Optional
Element:	**encryptedCardInfo**
Description:	Encrypted CardInfo. Contains of card information to be used on digitization process.
Type:	EncryptedPayload
Required:	Yes

SendPassCodeResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request returned by the Issuer.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Code: "00" for OK.
Type:	String
Required:	Yes
Element:	**processID**
Description:	Digitalization process identifier generated for each request returned by the Issuer.
Type:	String
Size:	Max 64
Required:	Yes
Element:	**messageDetail**
Description:	Detailed response message only for auditing purpose.
Type:	String
Required:	Optional
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

JSON Examples

SendPassCodeRequest

```
{
  "requestID": "4",
  "institutionCode": "HST",
  "processID": "1643ef957-622d-4137-abdf-fa605e81e72c",
  "tokenRefID": "DNITHE381502386342002358",
  "PANRefID": "V-3815023863409817870482",
  "authenticationMethod": "cell_phone",
  "OTP": "175824",
  "OTPExpiration":"2015-05-18 14:40:32”,
   "vaultIdentification": "VTS",
  "walletID": "N3GN-KWH6-NTYC-QNKN",
  "encryptedCardInfo": {
    "algorithm": "aes-ccm128",
    "nonce": "a96b3e84232d573c6592ceda",
    "encryptedData":
"KV1Mgkv40Nt4yggF1Ka7osdIkyMSsVe8K3o9wpQpMRTGeiXV2I65fIYgjZY1IGEpj/A7+KX3XB8C4
Foo8tEZ5xxQXa2PRudQ9B9s9WZbWoANcyaDAdw7ix7CQUN4x2ps9+oe8UaLtwjKrbKEDFkCML9rE9O
oco7vMr7y+uAlZ2NazPoWwx5fcQkn",
    "MACLength": 16
  }
}
```

Where:

//Plain CardInfo Object Data:

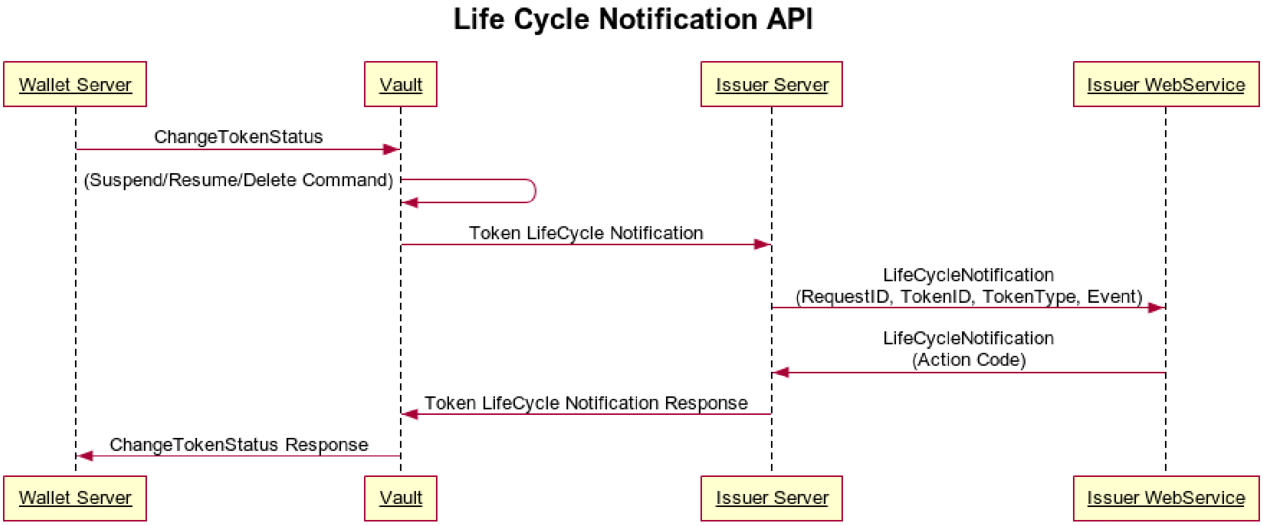
```
{
  "PAN": "1111110000000003",
  "expirationDate": {
    "month": "11",
    "year": "2024"
  },
  "CVV2": "500",
  "cardholderName": "FRANCISCO PEREIRA"
}
```

SendPassCodeResponse

```
{
  "requestID": "4",
  "returnCode": "00",
  "processID": "1643ef957-622d-4137-abdf-fa605e81e72c",
  "messageDetail": "Passcode received and sent to the user."
}
```

4.4. LifeCycleNotification

This API is used by Issuer server to send some notifications to Issuer to inform it about the life cycle status of tokens. As example, when a token is activated or deactivated this notification will be triggered.


Life Cycle Notification API
Wallet Server
Vault
Issuer Server
Issuer WebService
ChangeTokenStatus
(Suspend/Resume/Delete Command)
Token LifeCycle Notification
LifeCycleNotification
(RequestID, TokenID, TokenType, Event)
LifeCycleNotification
(Action Code)
Token LifeCycle Notification Response
ChangeTokenStatus Response
Wallet Server
Vault
Issuer Server
Issuer WebService

API endpoint	Method
Sandbox: https://{sandbox-issuer-host:port}/api/v3/lifecyclenotification	POST
Production: https://{issuer-host:port}/api/v3/lifecyclenotification	POST

LifeCycleNotificationRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes

Element:	**vaultIdentification**

Description:	Possible values are:
	“VTS” – for Visa;
	“MDES” – for Mastercard;
	“AMEX” – for Amex;
	“PL” – for Private Label.
	Used to identify the Vault in case tokenRefID does not exist in HST database.

Type:	String
Required:	Yes
Element:	**walletID**
Description:	Identifier of the wallet that generated the request. Some wallet providers such Apple Pay and Google pay define it with the same value used for device identification. For Apple Pay, this field shows the SEID.
Type:	String
Size:	64
Required:	Optional
Example:	Format at HST WhiteLabel Wallet: N3GN-KWH6-NTYC-QNKN
Element:	**tokenRequestorID**
Description:	Identification of the Token Requestor requesting digitization. It identifies SamsungPay, ApplePay, a Multi Issuer Wallet or an Issuer Wallet. All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project.
Type:	String
Size:	64
Required:	Yes
Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data, it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PANRefID for further use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others.
Type:	String
Size:	64
Required:	Yes
Element:	**tokenType**
Description:	Possible values are: “HCE”, “SE”, ”COF”, ”ECOM”, ”QRCODE” (Case-Sensitive).

Type:	String
Required:	Required for “VTS” and “MDES” Not present for “AMEX”
Element:	**dateTime**
Description:	Format: yyyy-MM-ddTHH:mm:ss.SSS The value will be in GMT.
Type:	String
Required:	Yes
Element:	**event**
Description:	Possible values are: “ACTIVATED” – When the token is activated by the vault , “SUSPENDED” – When the token is suspended by the vault, “CANCELLED” – When the token is cancelled by the vault, “INACTIVE” – When the token is inactive, provisioned in yellow flow and now requires further authentication of the cardholder. “DEVICE_BINDING_RESULT” – The token has been attempted to be bound on a trust device, “PENDING_ACTIVATION” – Alert triggered to the issuer every 24h notifying the token wasn’t activated yet (for Apple), “NO_FIRST_PURCHASE” – reserved for future use, “NO_RECENT_PURCHASE” – reserved for future use, “DELETED_FROM_CONSUMER_APP” – The token has been deleted from the consumer application. The token may still be active. (for MDES) “REDIGITIZATION_COMPLETE” – The token has been re-digitized to the device (for MDES)
Type:	String
Required:	Yes
Element:	**tokenUserInfo**
Description:	The information of the user that request the device binding.
Type:	tokenUserInfo Object
Size:	1
Required:	Optional for “VTS” and “MDES” Not present for “AMEX”
Element:	**merchantInfo**
Description:	The information of the merchant that request the device binding.
Type:	merchantInfo Object
Size:	1
Required:	Optional for “VTS” and “MDES” Not present for “AMEX”
Element:	**deviceBindingResult**
Description:	The possible values are: **“DEVICE_BINDING_APPROVED”** – Approved by green flow. **“DEVICE_BINDING_OTP”** – Approved by yellow flow through OTP method.

	"DEVICE_BINDING_CALL_CENTER" – Approved by yellow flow through Call Center method. **"DEVICE_BINDING_ISSUER_APP"** – Approved by yellow flow through App to App method. **"DEVICE_BINDING_REMOVED"** – The binding between the token and the device was removed.
Type:	String
Required:	Optional for "VTS" Not present for "MDES" and "AMEX"
Element:	**deviceInfo**
Description:	Information about the device associated to the token.
Type:	DeviceInfo
Required:	Optional
Element:	**encryptedCardInfo**
Description:	Encrypted CardInfo. Contains card information related to this notification.
Type:	EncryptedPayload
Required:	Optional
Element:	**encryptedTokenInfo**
Description:	Encrypted TokenInfo. Contains token information related to this
Type:	notification.
Required:	EncryptedPayload
	Optional
Element:	**processID**
Description:	Digitalization process identifier generated for each request.
Type:	String
Size:	Max 64
Required:	Optional.

LifeCycleNotificationResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request returned by the Issuer.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Code: "00" for OK.
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

JSON Examples

LifeCycleNotificationRequest

```
{
  "requestID": "4",
  "institutionCode": "HST",
  "vaultIdentification": "VTS",
  "walletID": "N3GN-KWH6-NTYC-QNKN",
  "tokenRequestorID": "42301999123",
  "tokenRefID": "DNITHE381502386342002358",
  "tokenType": "HCE",
  "dateTime": "2015-05-18T14:40:32.000Z",
  "event": "ACTIVATED"
}
```

LifeCycleNotificationResponse

```
{
  "requestID": "4",
  "returnCode": "00"
}
```

4.5. DeviceBindingEligibility

Device Binding consists of associating a device to an E-Commerce/COF Token. It is an additional layered security with trusted device management. Before initiating the device binding process, it is required by the TR-TSP to complete the enroll process of that device.

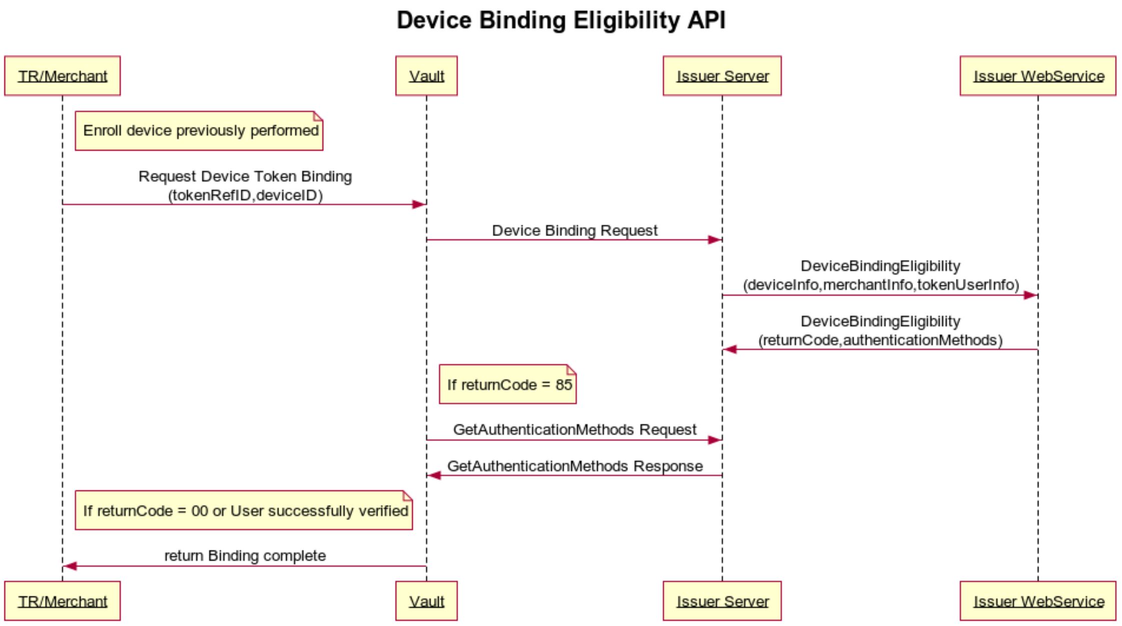
A token can be bound up to 100 devices. To distinguish them, the Vault generates a device index in the moment of the device binding.

To complete the device binding process, the Issuer must indicate if the user must or not be verified (green or yellow flow), according to the rules below:

- Yellow Flow: The Issuer must return the value **"85" – Device is eligible to be bound for this token and cardholder must be verified** in the *returnCode* element to present to the customer the ID&V methods available for identity validation. Cardholder will receive one or more options to choose (*Call Center, App to App and OTP*) depending on Issuer implementation.
- Green flow: There is no customer ID&V. The Issuer must return the value **"00" – Device is eligible to be bound for this token** in the *returnCode* element.

During the cryptogram request in an E-COM payment flow, if the device is bound, it must be provided in this request the deviceID of the device that is bound to the token. Otherwise, the cryptogram validation will fail.

This API is only used for VISA implementations.


Device Binding Eligibility API
TR/Merchant
Vault
Issuer Server
Issuer WebService
Enroll device previously performed
Request Device Token Binding (tokenRefID,deviceID)
Device Binding Request
DeviceBindingEligibility (deviceInfo,merchantInfo,tokenUserInfo)
DeviceBindingEligibility (returnCode,authenticationMethods)
If returnCode = 85
GetAuthenticationMethods Request
GetAuthenticationMethods Response
If returnCode = 00 or User successfully verified
return Binding complete
TR/Merchant
Vault
Issuer Server
Issuer WebService

API endpoint	Method
Sandbox: https://{sandbox-issuer-host:port}/api/v3/devicebindingeligibility	POST
Production: https://{issuer-host:port}/api/v3/devicebindingeligibility	POST

DeviceBindingEligibilityRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Size:	64
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes

Element:	**vaultIdentification**
Description:	Possible values are: “VTS” – for Visa; “MDES” – for Mastercard; “PL” – for Private Label. Used to identify the Vault in case tokenRefID does not exist in HST database.
Type:	String
Size:	32
Required:	Yes
Element:	**walletID**
Description:	Identifier of the wallet that generated the request. Some wallet providers such Apple Pay and Google pay define it with the same value used for device identification. For Apple Pay, this field shows the SEID.
Type:	String
Size:	64
Required:	Optional
Example:	Format at HST WhiteLabel Wallet: N3GN-KWH6-NTYC-QNKN
Element:	**tokenRequestorID**
Description:	Identification of the Token Requestor requesting digitization. It identifies SamsungPay, ApplePay, a Multi Issuer Wallet or an Issuer Wallet. All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project.
Type:	String
Size:	64
Required:	Yes
Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data, it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PANRefID for further use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others.
Type:	String
Size:	64
Required:	Yes
Element:	**PANRefID**
Description:	The PANRefID is a value assigned by the vault to identify the PAN. It identifies the PAN on the Vault.

	For VISA, each PAN generates a PANRefID value, which means a VISA PAN must have only one PANRefID value assigned. By using this data, it is not necessary to input the real PAN value. It is recommended to relate the PANRefID value with the TokenRefID for further use in APIs such as GetAssociatedTokens, GetPANByPANRefID and others.
Type:	String
Size:	64
Required:	Yes
Element:	**deviceInfo**
Description:	Data associated with the device. At least, deviceIndex and deviceID will be provided.
Type:	DeviceInfo Object
Required:	Yes
Element:	**tokenUserInfo**
Description:	The information of the user that request the device binding.
Type:	TokenUserInfo Object
Required:	Optional
Element:	**merchantInfo**
Description:	The information of the merchant that request the device binding.
Type:	MerchantInfo Object
Required:	Optional
Element:	**tokenInfo**
Description:	Encrypted TokenInfo of to the token created related to the card being digitized.
Type:	EncryptedPayload
Required:	Optional

DeviceBindingEligibilityResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request returned by the Issuer.
Type:	String
Size:	64
Required:	Yes
Element:	**returnCode**
Description:	Possible values are: **“00”** – Device is eligible to be bound for this token (green flow); **“05”** – Device is not eligible to be bound for this token; **“85”** – Device is eligible to be bound for this token and cardholder must be verified (yellow flow).
Type:	String
Required:	Yes

Element:	**authenticationMethods**
Description:	Authentication methods list for specific user’s device, if authentication needed. The possible values for implementation are OTP, Call Center and App-to-App and their details are described on the AuthenticationMethod description element.
Type:	Array <AuthenticationMethod>
Required:	Optional, only if returnCode element returns value “85”

JSON Examples

DeviceBindingEligibilityRequest

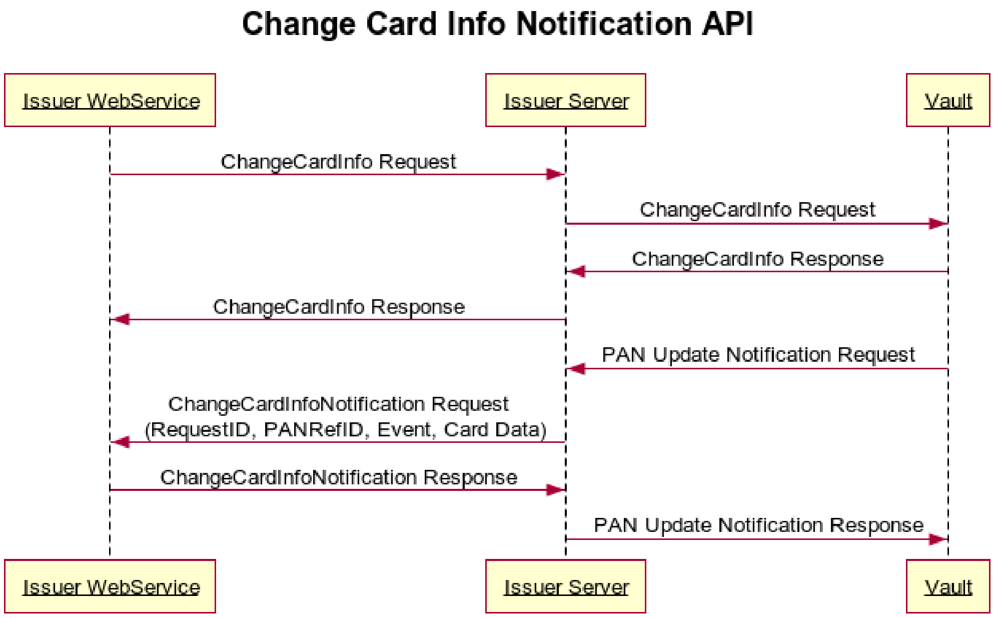
```
{
  "requestID": "5",
  "institutionCode": "HST",
  "vaultIdentification": "VTS",
  "walletID": "N3GN-KWH6-NTYC-QNKN",
  "tokenRequestorID": "42301999123",
  "tokenRefID": "DNITHE381502386342002358",
  "PANRefID": "V-3815023863409817870482",
  "deviceInfo": {
    "deviceType": "MOBILE_PHONE",
    "deviceNumber": "5355",
    "deviceName": "Mary’s Phone",
    "serialNumber": "16344-536536-5453",
    "deviceID": "1234556675587",
    "deviceIndex": "02"
  },
  "tokenUserInfo": {
    "ID": "98765679864",
    "appType": "MOBILE_WEB"
  },
  "merchantInfo": {
    "ID": "12345678",
    "merchantName": "ABC STORE"
  }
}
```

DeviceBindingEligibilityResponse

```
{
  "requestID": "5",
  "returnCode": "85",
  "authenticationMethods": [
    {
      "identifier": "125485644",
      "type": "bank_app",
      "maskedInfo": "Mobile Banking App",
      "sourceAddress": "com.DemoBank.DemoApp",
      "platform": "ANDROID"
    },
    {
      "identifier": "125485644",
      "type": "cell_phone",
      "maskedInfo": "XXX-XXX-1234"
    }
  ]
}
```

4.6. ChangeCardInfoNotification

This API is used by Issuer server to send some notifications to Issuer whenever it is performed PAN or PAN expiration date updates. This API is used only in VTS.


Change Card Info Notification API
Issuer WebService
Issuer Server
Vault
ChangeCardInfo Request
ChangeCardInfo Request
ChangeCardInfo Response
ChangeCardInfo Response
PAN Update Notification Request
ChangeCardInfoNotification Request
(RequestID, PANRefID, Event, Card Data)
ChangeCardInfoNotification Response
PAN Update Notification Response
Issuer WebService
Issuer Server
Vault

API endpoint	Method
Sandbox: https://{sandbox-issuer-host:port}/api/v3/changecardinfonotification	POST
Production: https://{issuer-host:port}/api/v3/changecardinfonotification	POST

ChangeCardInfoNotificationRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes
Element:	**vaultIdentification**
Description:	Possible values are: “VTS” – for Visa; Used to identify the Vault. This API is used only by VTS.
Type:	String
Required:	Yes
Element:	**PANRefID**
Description:	The PANRefID is a value assigned by the vault to identify the PAN. It identifies the PAN on the Vault. For VISA, each PAN generates a PANRefID value, which means a VISA PAN must have only one PANRefID value assigned. For Mastercard, the PANRefID it is associated to the Token Requestor, which means it is not unique for a PAN and it can have multiples PANRefIDs. By using this data, it is not necessary to input the real PAN value. It is recommended to relate the PANRefID value with the TokenRefID for further use in APIs such as GetAssociatedTokens, GetPANByPANRefID and others.
Type:	String
Size:	64
Required:	Optional
Element:	**dateTime**
Description:	Format: yyyy-MM-ddTHH:mm:ss.SSSZ The value will be in GMT.
Type:	String
Required:	Yes
Element:	**event**
Description:	Possible values are: “PAN_UPDATED” (Case-Sensitive).
Type:	String
Required:	Yes

Element:	**messageReasonType**
Description:	Possible values are: "ACCOUNT_UPDATE" or "EXP_DATE_UPDATE" (Case-Sensitive).
Type:	String
Required:	Yes
Element:	**encryptedOldCardInfo**
Description:	CardInfo - Old encrypted card information, containing the current PAN and expiration date.
Type:	EncryptedPayload
Required:	Yes
Element:	**encryptedNewCardInfo**
Description:	CardInfo - New encrypted card information, containing the new PAN and expiration date.
Type:	EncryptedPayload
Required:	Yes

ChangeCardInfoNotificationResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request returned by the Issuer.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Code: “00” for OK.
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

JSON Examples

ChangeCardInfoNotificationRequest

```
{
  "requestID": "4",
  "institutionCode": "HST",
  "vaultIdentification": "VTS",
  "PANRefID": "V-3815023863409817870482",
  "dateTime": "2015-05-18T14:40:32.000Z",
  "event": "PAN_UPDATED",
  "messageReasonType": "EXP_DATE_UPDATED",
  "encryptedOldCardInfo": {
    "algorithm": "aes-gcm256",
    "iv": "228be5ada04ab22ae2834fba3f1be459",
    "encryptedData":
"j6RlcievkUE+LQOusfSOfLDaYt99wnVsfCih9G1190ChD74Zewum6337f+V2WeVcAZjFPm9UZlB3E
0dpORKFWlFvsYXfjalTvlY+4X48ie0mIMx5MnLoIg==",
    "MACLength": 16
  },
  "encryptedNewCardInfo": {
    "algorithm": "aes-gcm256",
    "iv": "e434a9e356425c86338c91bd",
    "encryptedData":
"/rkGCXbH5kibl+hFOa5sMUZV5yckICCs/GT0EkTpFQcJ8xo0/1GBcEQC/vK2UsOBQ/qgILi2I3SOo
NRI5XwNPRg33VjehErWBVjv42nGSUc1NxyhvglpOQ==",
    "MACLength": 16
  }
}
```

Where:

//Plain OldCardInfo Object Data:

```
{
  "PAN": "1111110000001234",
  "expirationDate": {
    "month": "08",
    "year": "2025"
  }
}
```

//Plain NewCardInfo Object Data:

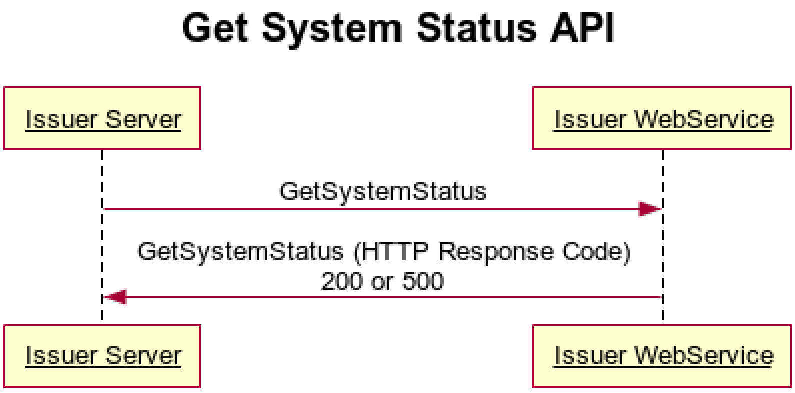
```
{
  "PAN": "1111110000004321",
  "expirationDate": {
    "month": "05",
    "year": "2026"
  }
}
```

ChangeCardInfoNotificationResponse

```
{
  "requestID": "4",
  "returnCode": "00"
}
```

4.7. GetSystemStatus

This API is used by Issuer Server to monitor and to check Issuer system's health status. It is recommended to be one of the first APIs to be implemented during a I-TSP/TR-TSP project to establish and validate a connection between Issuer and HST systems.



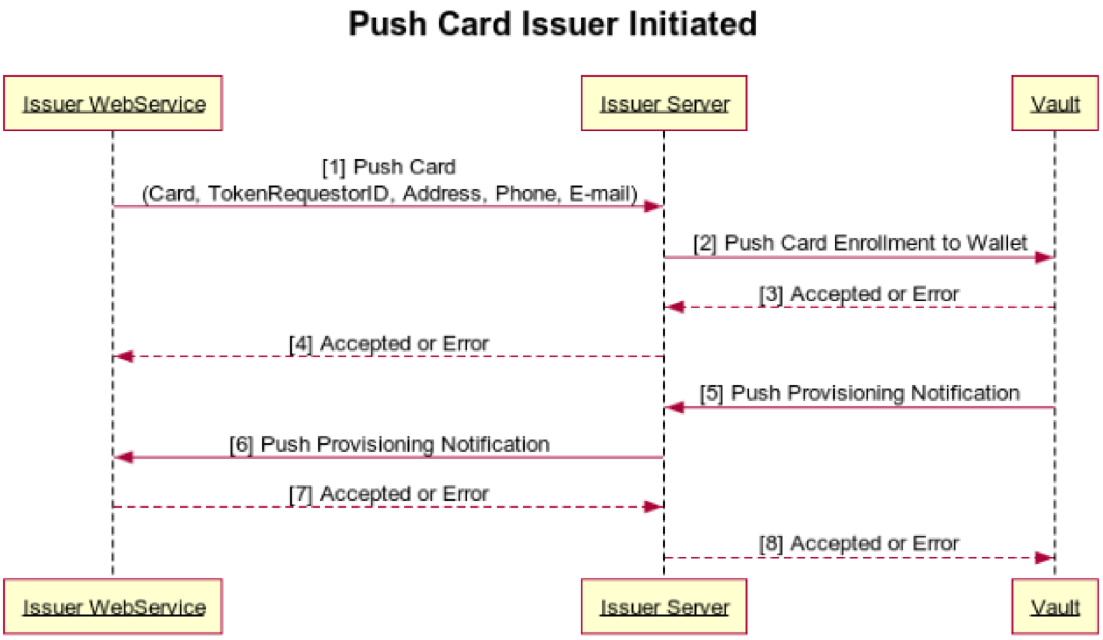
API endpoint	Method
Sandbox: https://{sandbox-issuer-host:port}/api/v3/getsystemstatus	GET
Production: https://{issuer-host:port}/api/v3/getsystemstatus	GET

Issuer should respond with 200 if OK or 5XX in case of error or unavailability.

4.8. PushProvisioningNotification

This API is used by Issuer Server to send notifications to Issuer about the updating of the push provisioning status.

This API is only used for VISA implementations. The notification is per wallet provider per provisioning action. If the issuer pushes a payment instrument to multiple wallet providers, it will receive multiple notifications for that payment instrument.


Push Card Issuer Initiated
Issuer WebService
Issuer Server
Vault
[1] Push Card
(Card, TokenRequestorID, Address, Phone, E-mail)
[2] Push Card Enrollment to Wallet
[3] Accepted or Error
[4] Accepted or Error
[5] Push Provisioning Notification
[6] Push Provisioning Notification
[7] Accepted or Error
[8] Accepted or Error
Issuer WebService
Issuer Server
Vault

API endpoint	Method
Sandbox: https://{sandbox-issuer-host:port}/api/v3/pushprovisioningnotification	POST
Production: https://{issuer-host:port}/api/v3/pushprovisioningnotification	POST

PushProvisioningNotificationRequest

Element:	**tokenRequestorID**
Description:	Identification of the Token Requestor. It identifies ClickToPay, SamsungPay, ApplePay, a Multi Issuer Wallet, an Issuer Wallet or a Merchant.. All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project.
Type:	String
Size:	64
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes
Element:	**event**
Description:	Provisioning action status. Possible values are: “SUCCESS” – Token is provisioned successfully “NOTIFICATION_FAILURE” – Failed to send push provision notification to the wallet provider (token requestor) “PROVISION_FAILURE” – Failed to provision the token
Type:	String
Required:	Yes
Element:	**encryptedPushNotification**
Description:	PushNotification. Contains of cardholder information provided by the vault.
Type:	EncryptedPayload
Required:	Yes

PushProvisioningNotificationResponse

Element:	**returnCode**
Description:	Return Code: “00” for OK.
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

JSON Examples

PushProvisioningNotificationRequest

```
{
  "tokenRequestorID": "40010075338",
  "event": "SUCCESS",
  "encryptedClientInformation": {
    "algorithm": "aes-gcm256",
    "iv": "2415F6220825A8BC7B7A47233F46C378",
    "encryptedData": "GK5NfIXesgJ8loyzqKOJh4Zhg7Lbf3fzsVre43iU3F4qRv1zGTI
seLteLYHUMNze1gTO186aPzMPMlOuL4f3S3CI7b0bzOcmfxadk2hVq6/A",
        "MACLength": 12
  }
}
```

Where:
//Plain PushNotification Object Data

```
{
  "source": "ISSUER",
  "firstName": "ClientFristName",
  "middleName": "ClientMiddleName",
  "lastName": "ClientLastName",
  "contactPhone": "+44791112345",
  "contactEmail": "name@mail.com",
  "locale": "en_US",
  "deviceID": "...",
  "tokenRefID": "..."
}
```

PushProvisioningNotificationResponse

```
{
  "returnCode": "00"
}
```

NOTE: In error case, the response is:

```
{
  "returnCode": "98",
  "errorDescription": "Invalid Request"
}
```

4.9. BulkProvisionNotification

This API is used by Issuer Server to send notifications to Issuer about the updating of the bulk push provision operations.

API endpoint	Method
Sandbox: https://{sandbox-issuer-host:port}/api/v3/bulkprovisionnotification	POST
Production: https://{issuer-host:port}/api/v3/bulkprovisionnotification	POST

BulkProvisionNotificationRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**institutionCode**
	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes
Element:	**bulkPushReceiptID**
Description:	A receipt to associated to the Bulk Provision request.
Type:	String
Size:	36
Required:	Yes
Element:	**event**
Description:	Provisioning action status. Possible values are: “JOB_FINISHED” – The bulk provision job was successfully processed. “JOB_FAILED” – The bulk provision job was processed with errors. “JOB_PENDING” – The bulk provision job didn’t finish. String
Type:	Yes
Required:	

BulkProvisionNotificationResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Code: “00” for OK.
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

5. Tokenization BUS - Inbound

The HST Tokenization BUS webservice is designed to allow issuers to integrate its current CMS (Card Management System) or Internet Banking directly with the Issuer server. In such way, it is possible to perform a series of operations within its own platform.

5.1. GetAssociatedTokens

This API is used to get the Token Reference IDs associated to a PAN, PAN Reference ID and/or UserID. Then, it is necessary to call GetTokenInfo to obtain details about the token. For Issuer Wallets it’s also possible to search for tokens associated to an UserID previously defined on GetAvailableCards or AuthenticateCardholder (Issuer Wallet APIs described in other documentation).

1-) In case PAN and also PAN Reference ID elements were both sent during request, only the PAN Reference ID will be used, and PAN element will be ignored.

2-) If the request is performed using PAN element, the results can return **all** the tokensReferenceIDs associated to the cardholder, regardless the device.

3-) If the request is performed using PANRefID element, only the tokens associated to such PANRefID will be returned. This PANRefID will be related to a unique device, i.e., it can be not related to all the tokens associated to the cardholder.

4-) Since the UserID is an information not available in the Vault, if this element is used in the search, the inquiry will be performed **only** in the HST Environment local database, for associated tokens retrieving. In this case the Issuer Server will not send the request to the Vault(s).

5-) If multiple elements are provided in the request message, the priority order used during the search will be (from the highest to lowest): userID, PANRefID and cyphered PAN elements, respectively.

API endpoint	**Method**
Sandbox: https://issuer-bus.test-teste-prueba.com:9215/api/v3/getassociatedtokens	POST
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/getassociatedtokens	POST

GetAssociatedTokensRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Required:	Yes
Element:	**tokenRequestorID**
Description:	The token requestor associated with the token. It identifies SamsungPay, ApplePay, a Multi Issuer Wallet, an Issuer Wallet or a Merchant. All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project. If provided, results will only contain tokens related to that specific Token Requestor ID.
Type:	String
Required:	Optional

Element: **tokenType**
Description: Results will only contain tokens of the specified type.
Possible values are: “HCE”, “SE”, “COF”, “ECOM”, “QRCODE” (Case-Sensitive).
Type: String
Size: 32
Required: Optional

Element: **userID**
Description: Issuer identification of the cardholder. Typically, an account or online banking user ID defined on response of GetAvailableCards or AuthenticateCardholder.
Only for auditing purpose on HST’s system, there is no participation during the provisioning and transaction flows.
Type: String
Required: Optional

Element: **PANRefID**
Description: The PANRefID is a value assigned by the vault to identify the PAN. It identifies the PAN on the Vault.
For VISA, each PAN generates a PANRefID value, which means a VISA PAN must have only one PANRefID value assigned.
For Mastercard, the PANRefID it is associated to the Token Requestor, which means it is not unique for a PAN and it can have multiples PANRefIDs.
By using this data, it is not necessary to input the real PAN value. It is recommended to relate the PANRefID value with the TokenRefID for further use in APIs such as GetAssociatedTokens, GetPANByPANRefID and others.
Type: String
Size: 64
Required: Optional

Element: **encryptedCardInfo**
Description: Contains a PAN number on an encrypted CardInfo object. PAN is the only attribute of CardInfo that must be populated.
Type: EncryptedPayload
Required: Optional for “VTS” and “MDES”
Required for “AMEX”

Element: **tokenState**
Description: Searches for tokens in a specific state.
Possible values are: “ACTIVE”, “SUSPENDED”, “INACTIVE”, “CANCELED” (Case-Sensitive).
Type: String
Size 0-32
Required: Optional for “VTS” and “MDES”
Not present for “AMEX”

Element: **operatorID**

Description:	The operator identification code.
Type:	String
Size:	0-16
Required:	Required for VTS and MDES. Not present otherwise.
Element:	**operatorName**
Description:	Operator name.
Type:	String
Size:	0-200
Required:	Required for MDES. Not present otherwise.
Element:	**operatorPhone**
Description:	Operator’s contact phone.
Type:	String
Size:	0-20
Required:	Required for MDES. Not present otherwise.
Element:	**vaultIdentification**

Description:	Possible values are:
	“VTS” – for Visa;
	“MDES” – for Mastercard;
	“AMEX” – for Amex;
	“PL” – for Private Label.
	Used to identify the Vault in case the tokenRefID does not exist in HST database.

Type:	String
Size:	32
Required:	Required for “AMEX”
Element:	**cardKey**
Description:	ID of the internal Amex card, in case the issuer has this data
Type:	String
Size	64
Required:	Optional for “AMEX” Not present for “VTS” and “MDES”

GetAssociatedTokensResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Code: “00” for OK. Return Code: “01” for Ok with a warning condition – Check Error description for more information.
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional
Element:	**tokenRefIDList**
Description:	List of Token Reference IDs.
Type:	Array <String>
Required:	Optional
Element:	**tokenInfoList**
Description:	List of encrypted TokenInfo objects. This list has paired indexes with tokenRefIDList elements.
Type:	Array <EncryptedPayload>
Required:	Optional

JSON Examples

GetAssociatedTokensRequest

```
{
  "requestID": "9",
  "institutionCode": "HST",
  "tokenRequestorID": "42301999123",
  "tokenType": "HCE",
  "PANRefID": "V-3815023863409817870482",
  "encryptedCardInfo": {
    "algorithm": "aes-ccm128",
    "nonce": "a96b3e84232d573c6592ceda",
    "encryptedData":
"KV1Mgkv40Nt4yggF1Ka7osdIkyMSsVe8K3o9wpQpMRTGeiXV2I65fIYgjZY1IGEpj/A7+KX3XB8C4
Foo8tEZ5xxQXa2PRudQ9B9s9WZbWoANcyaDAdw7ix7CQUN4x2ps9+oe8UaLtwjKrbKEDFkCML9rE9O
oco7vMr7y+uAlZ2NazPoWwx5fcQkn",
    "MACLength": 16
  },
  "tokenState": "INACTIVE"
}
```

Where:

//Plain CardInfo Object Data:

```
{
  "PAN": "1111110000000003",
  "expirationDate": {
    "month": "11",
    "year": "2024"
  },
  "CVV2": "500",
  "cardholderName": "FRANCISCO PEREIRA"
}
```

GetAssociatedTokensResponse

```
{
  "requestID": "9",
  "returnCode": "00",
  "tokenRefIDList": [
    "DNITHE381502386342002358",
    "A4N6HKA45114456AS4584844"
  ],
  "tokenInfoList": [
    {
      "algorithm": "aes-gcm256",
      "iv": "F6721F7B3A63A8F4908CF5245B154120",
      "encryptedData": "**********...",
      "MACLength": 12
    },
    {
      "algorithm": "aes-gcm256",
      "iv": "ECAE3F12E0E73177A030084B265EE055",
      "encryptedData": "**********...",
      "MACLength": 12
    }
  ]
}
```

Where:

//Plain TokenInfo Object Data:

```
  {
    "token": "1111113245678979",
    "expirationDate": {
      "month": "10",
      "year": "2024"
    },
    "state": "ACTIVE",
    "type": "HCE",
    "lastTokenStatusUpdatedTimeStamp": "2015-05-18T14:40:32.000Z",
    "entityOfLastAction": "ISSUER",
    "deviceInfo": {
      "deviceType": "MOBILE_PHONE",
      "deviceNumber": "1234",
      "deviceName": "AndroidCellPhone",
      "serialNumber": "874759678487"
    },
    "OTPCodeIndicate": "PRESENT",
    "OTPCodeExpiration": "2015-05-18T14:40:32.000Z",
    "PANsLastFour": "1234",
    "previousPANsLastFour": "4653",
    "tokenRequestorID": "42301999123",
    "tokenRefID": "DNITHE381502386342002358"
  }
```

5.2. GetTokenInfo

This API is used to get all the information about a token. During request, the tokenRefID must be sent.

For Mastercard, the token value will not be provided because the brand doesn't provide this value in this API. The only API the issuer may receive the token is DigitizationNotification.

API endpoint	Method
Sandbox: https://issuer-bus.test-teste-prueba.com:9215/api/v3/gettokeninfo	POST
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/gettokeninfo	POST

GetTokenInfoRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes

Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Required:	Yes

Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data, it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PANRefID for further use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others.
Type:	String
Size	64
Required:	Required for “MDES”, “VTS” and “AMEX”

Element:	**deviceBindingInfo**
Description:	True if it must return device binding data or false if not. By default, it is False. If True, deviceInfo Object list must return on the response deviceIDs and deviceIndexes bound to the Token.
Type:	Boolean
Required:	Used only for “VTS” Not present for “MDES” and “AMEX”

Element:	**vaultIdentification**

Description:	Possible values are:
	“VTS” – for Visa;
	“MDES” – for Mastercard;
	“AMEX” – for Amex;
	“PL” – for Private Label.
	Used to identify the Vault in case the tokenRefID does not exist in HST database.

Type:	String
Size:	32
Required:	Required for “AMEX” Optional for “VTS” and “MDES”

Element:	**operatorID**
Description:	The operator identification code.
Type:	String
Size:	0-16
Required:	Required for VTS and MDES. Not present otherwise.

Element:	**operatorName**
Description:	Operator name.
Type:	String
Size:	0-200
Required:	Required for MDES. Not present otherwise.

Element:	**operatorPhone**
Description:	Operator contact phone.
Type:	String
Size:	0-20
Required:	Required for MDES. Not present otherwise.

Element:	**tokenRequestorID**
Description:	The token requestor associated with the token. It identifies SamsungPay, ApplePay, a Multi Issuer Wallet, an Issuer Wallet or a Merchant. All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project.
Type:	String
Size:	64
Required:	Required for VTS, Optional for MDES

GetTokenInfoResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes

Element:	**returnCode**
Description:	Return Code: “00” for OK “92” for Token Not Found “95” for Cryptography Error “96” for Invalid Data “97” for Required Data Missing “98” for Invalid Request “99” for System Error, please check error description.
Type:	String
Required:	Yes

Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional
Element:	**tokenInfo**
Description:	Encrypted list of TokenInfo objects related to the requested tokenRefIDs.
Type:	EncryptedPayload
Required:	Optional
Element:	**deviceInfo**
Description:	List of DeviceInfo Objects related to the requested tokenRefID. It is returned if deviceBindingInfo element in the request is True. The list of DeviceInfo objects will contain only the deviceID and deviceIndex elements.
Type:	List of deviceInfo Objects
Required:	Optional
Element:	**tokenRequestorID**
Description:	Identification of the Token Requestor associated to the token.
Type:	String
Required:	Optional
Element:	**RiskInformation**
Description:	RiskData provided by the Token Requestor on digitization process.
Type:	RiskInformation Object
Required:	Optional

JSON Examples

GetTokenInfoRequest

```
{
  "requestID": "4",
  "institutionCode": "HST",
  "tokenRefID": "DNITHE381502386342002358",
  "deviceBindingInfo": "True"
}
```

GetTokenInfoResponse

```
{
  "requestID": "4",
  "returnCode": "00",
  "tokenInfo": {
    "algorithm": "aes-ccm128",
    "nonce": "b3c0f84e500e50ffcd5f563e",
    "encryptedData":
"Q6sfnucc1f6duTMvzcUa5SueAKUeDpd2Fq+fcSg/xBFU0LhSoiTMJ/3BiZc6uP5GrWbUouoSr01ve
r9YiauDloy9hD4buW2ZiE24sguOpjhlsx2DyNX0ryBlJOjyhK/9z9dfQaRSwK6TxBmndsMAOCGRf5g
QiwiFdgF7w/xcJfoDrSnQ9MPkLThyIAA7+y+8ZLiFjjRJGAY1fXjoNnVjsDsxPuIq+p5hI0BrQ9YWH
CqCllbDX5PycBMT7e5jL2dgz4p7hP2fNrlmXY5EVqhPD12FbjSliXKNib4RdJe/xbol5WCzwhsxncu
+8Owt0VMzdZs6DdcrDcMMmB4l+5UAsrzx73JhkAhO0j5NK2u+llrwrAcn8Ul+A/tFv1W3HrarixA1X
PLVpGdOq+3DgjxqkLBZOV1WiZ0D+q0vtVrmkqUvvlyzZafcLufMw9/7KX1sONmvQDP+2zC1R96VghQ
Njj3wIo7xH/+T0TKhUMqwCapvxkSwD70l87z/eYPKmIb4YXWgbiyKnRUyhCnE5vDxYAlOt8+5mz0LY
nJtLAPEMvtyxmIsFU6GW+AYvVJb3ae9ZNfcdsK9DkHpEmHIQ0UffvEAv7ELgjZALWOV1AsxlHiBLJd
YxGXO+3BPuUJssFc1P99AXWyKOTY51KBJMVsWxHc=",
    "MACLength": 16
  },
  "tokenRequestorID": "42301999123",
  "deviceInfo": [
    {
      "deviceID": "87755776656",
      "deviceIndex": "01"
    }
  ]
}
```

Where:

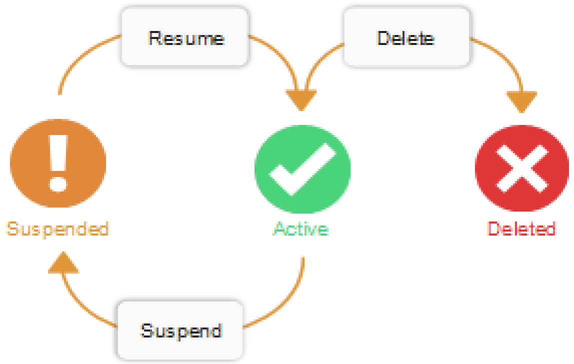
//Plain TokenInfo Object Data:

```
[
  {
    "token": "1111113245678979",
    "expirationDate": {
      "month": "10",
      "year": "2024"
    },
    "state": "ACTIVE",
    "type": "HCE",
    "lastTokenStatusUpdatedTimeStamp": "2015-05-18T14:40:32.000Z",
    "entityOfLastAction": "ISSUER",
    "deviceInfo": {
      "deviceType": "MOBILE_PHONE",
      "deviceNumber": "1234",
      "deviceName": "AndroidCellPhone",
      "serialNumber": "874759678487"
    },
    "OTPCodeIndicate": "PRESENT",
    "OTPCodeExpiration": "2015-05-18T14:40:32.000Z",
    "PANsLastFour": "1234",
    "previousPANsLastFour": "4653",
    "tokenRefID": "DNITHE381502386342002358"
  }
]
```

5.3. ChangeTokenStatus

This API is used to change the status of a token. Issuer must inform the Token Reference ID to perform the operation. Through this API, it is possible to activate, suspend, resume and delete a token. The conditions are described below:

- A token can be **activated** from inactive status after a cardholder verification is performed by the Issuer.
- A token may be **suspended** because of a stolen/lost device or card. Once is submitted with a suspension reason, the status is changed to “suspended” and the token can no longer be used for payments unless it is activated again.
- A token can be **reactivated (resumed)** from a suspension after the cardholder recovers a lost device/card and request the activation to the Issuer.
- Any token can be **deleted** due cardholder reasons (*lost/stolen card or device, closed PAN, etc.*), regardless the actual token status. Once a token is deleted, it can no longer be used for payments or activated again (*).



Besides token status lifecycle, this API can also be used to manage the ***Device binding*** lifecycle, allowing token **device binding approval and removal operations**.

(*) Please be aware that VTS will provision the same tokenRefID for a HCE token when occurs within seven days from the token deletion by the cardholder. It doesn’t mean it is possible to revert a deleted token from lifecycle operations.

API endpoint	Method

Sandbox: https://issuer-bus.test-teste-prueba.com:9215/api/v3/changetokenstatus	POST
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/changetokenstatus	POST

ChangeTokenStatusRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Required:	Yes
Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data, it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PANRefID for further use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others. Contains the Token Reference ID that is subject to the status change.
Type:	String
Size:	64
Required:	Yes
Element:	**action**
Description:	Possible values are: “DELETE”, “SUSPEND”, “RESUME”, “DEVICE_BINDING_APPROVE”, “DEVICE_BINDING_REMOVE”.
Type:	String
Required:	Yes
Element:	**operatorID**
Description:	The operator identification code.
Type:	String
Size:	0-16
Required:	Required for VTS and MDES. Not present otherwise.

Element:	**operatorName**
Description:	Operator name.
Type:	String
Size:	0-200
Required:	Required for MDES. Not present otherwise.
Element:	**operatorPhone**
Description:	Operator contact phone.
Type:	String
Size:	0-20
Required:	Required for MDES. Not present otherwise.
Element:	**reason**
Description:	The reason why the change was made.
Type:	String
Size:	256
Required:	Required for “VTS” and “MDES” Not present for “AMEX”
Element:	**additionalInformation**
Description:	During the change token status process, it is possible to add more information if desired by the helpdesk operator in order to complement the reason already indicated.
Type:	String
Size:	0-256
Required:	Optional
Element:	**vaultIdentification**

Description:	Possible values are:
	“VTS” – for Visa;
	“MDES” – for Mastercard;
	“AMEX” – for Amex;
	“PL” – for Private Label.
	Used to identify the Vault in case the tokenRefID does not exist in HST database.

	String
Type:	32
Size:	Optional for “VTS” and “MDES”
Required:	Required for “AMEX”
Element:	**deviceInfo**
Description:	Only valid for *Device Binding* lifecycle operations. Only deviceID and deviceIndex must be informed.
Type:	DeviceInfo Object
Required:	Optional para “VTS” Not present for “MDES” and “AMEX”
Element:	**tokenRequestorID**
Description:	Identification of the Token Requestor requesting digitization. It identifies SamsungPay, ApplePay, a Multi Issuer Wallet or an Issuer Wallet.

	All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project.
Type:	String
Size:	64
Required:	Required for “VTS”. Optional for “MDES” Not present for “AMEX”

ChangeTokenStatusResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Code: “00” for OK. “91” for Invalid Token Status/Token Not Active “92” for Token Not Found “93” for Token Already in the State Requested “95” for Cryptography Error “96” for Invalid Data “97” for Required Data Missing “98” for Invalid Request “99” for System Error, please check error description
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

JSON Examples

ChangeTokenStatusRequest

```
{
  "requestID": "4",
  "institutionCode": "HST",
  "tokenRefID": "DNITHE381502386342002358",
  "action": "DELETE",
  "operatorID": "134",
  "reason": "About to expire"
}
```

ChangeTokenStatusResponse

```
{
  "requestID": "4",
  "returnCode": "00"
}
```

5.4. ActivateToken

This API is used to activate a token informing the Token Reference ID during ID&V flow (call center inbound call, call center outbound call or App to App).

Mostly used to activate a token on a digitalization flow that requires cardholder identification and verification (ID&V) with the authentication method App-to-App (if used by the Issuer) during the yellow flow. Also can be invoked by Issuer Card Management tools (Helpdesk).

API endpoint	Method
Sandbox: https://issuer-bus.test-teste-prueba.com:9215/api/v3/activatetoken	POST
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/activatetoken	POST

ActivateTokenRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	

Required:	String Yes
Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data, it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PANRefID for further use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others. Token Reference ID associated to the token being activated. This element must be sent when the cardholder decides to activate a single token.
Type:	String
Size:	64
Required:	Yes
~~Element:~~	~~**activationCode**~~
~~Description:~~	~~This can be a random code generated by the issuer only as an auditing purpose to be associated to the successful activation process and it is not validated by the Vault.~~
~~Type:~~	~~String~~
~~Size:~~	~~0-16~~
~~Required:~~	**Deprecated**
Element:	**operatorID**
Description:	The operator identification code.
Type:	String
Size:	0-16
Required:	Required for VTS and MDES. Not present otherwise.
Element:	**operatorName**
Description:	Operator name.
Type:	String
Size:	0-200
Required:	Required for MDES. Not present otherwise.
Element:	**operatorPhone**
Description:	Operator contact phone.
Type:	String
Size:	0-20
Required:	Required for MDES. Not present otherwise.
Element:	**reason**
Description:	The reason why the activation was made.
Type:	String
Required:	Yes

Element:	**vaultIdentification**

Description:	Possible values are:
	“VTS” – for Visa;
	“MDES” – for Mastercard;
	“AMEX” – for Amex;
	“PL” – for Private Label.
	Used to identify the Vault in case the tokenRefID does not exist in HST database.

Type:	String
Size:	32
Required:	Optional for “VTS” and “MDES” Required for “AMEX”
Element:	**tokenRequestorID**
Description:	The token requestor associated with the token. It identifies SamsungPay, ApplePay, a Multi Issuer Wallet, an Issuer Wallet or a Merchant.
Type:	String
Size:	64
Required:	Required

ActivateTokenResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Code: “00” for OK.
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

JSON Examples

ActivateTokenRequest

```
{
  "requestID": "4",
  "institutionCode": "HST",
  "tokenRefID": "DNITHE381502386342002358",
  "operatorID": "14",
  "reason": "Token activation pending"
}
```

ActivateTokenResponse

```
{
  "requestID": "4",
  "returnCode": "00"
}
```

5.5. GetPANByPANRefID

This API is a helper function exclusive for the App to App authentication. Issuers using this step-up method need to retrieve PAN information based on PAN Reference ID to verify if the card being digitized is related to the cardholder being authenticated on the Issuer app.

During App to App step-up method the Issuer receives a PANRefID value on its mobile application, and through this API the Issuer can get the PAN to validate if the cardholder has the PAN that is trying to digitize.

API endpoint	Method
Sandbox: https://issuer-bus.test-teste-prueba.com:9215/api/v3/getpanbypanrefid	POST
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/getpanbypanrefid	POST

GetPANByPANRefIDRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Required:	Yes

Element:	**PANRefID**
Description:	The PANRefID is a value assigned by the vault to identify the PAN. It identifies the PAN on the Vault. For VISA, each PAN generates a PANRefID value, which means a VISA PAN must have only one PANRefID value assigned. For Mastercard, the PANRefID it is associated to the Token Requestor, which means it is not unique for a PAN and it can have multiples PANRefIDs. By using this data, it is not necessary to input the real PAN value. It is recommended to relate the PANRefID value with the TokenRefID for further use in APIs such as GetAssociatedTokens, GetPANByPANRefID and others.
Type:	String
Required:	Required for “VTS” and “MDES” Not present for “AMEX”

GetPANByPANRefIDResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Code: “00” for OK.
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional
Element:	**encryptedPAN**
Description:	Encrypted PAN Number. Contains a string containing the card PAN related to the PAN Reference ID.
Type:	EncryptedPayload
Required:	Yes

JSON Examples

GetPANByPANRefIDRequest

```
{
  "requestID": "4",
  "institutionCode": "HST",
  "PANRefID": "V-3815023863409817870482"
}
```

GetPANByPANRefIDResponse

```
{
  "requestID": "4",
  "returnCode": "00",
  "encryptedPAN": {
    "algorithm": "aes-gcm256",
    "iv": "515B6D4BC91BDA4E8FFF1D5D246657AB",
    "encryptedData":
"8ZqX1V9oDgJfOUqHdam7nwtWgT595qDN+T1QFIGc4/Jzw6McJKW2FWsr",
    "MACLength": 16
  }
}
```

Where:

//Plain CardInfo Object Data:

```
{
  "PAN": "1111110000000003"
}
```

5.6. ChangeCardInfo

This API is used either to replace an old PAN for a new PAN in such a way that all existing tokens will be tied with the new PAN and the cardholder doesn't need to provision again.

After the process is executed, the Issuer will receive the new PAN when the user performs a transaction with the existing tokens.

Moreover, the Issuer can also extend the expiration date for a current card.

These are the use conditions for this API:

- To replace a PAN:
    - ➢ In the request message the PANs and expiration dates must be provided in both objects *encryptedOldCardInfo* and *encryptedNewCardInfo.*
    - ➢ **The new PAN must not have any associated tokens, i.e., the new card must not have been digitized yet in any other wallet or merchant**.
    - ➢ **Based on the previous condition, it's highly recommended the execution of this command before providing the new card to the cardholder or before the cardholder activate it.**

- **For Visa Cards, when replacing an existing PAN with a new one, if there are tokens associated with the existing PAN, the new PAN must have the same BIN as the original PAN. If the existing PAN and the replacement PAN have different BINs, you must delete the tokens on the existing PAN before calling the ChangeCardInfo API to replace the PAN.**

- To extend the expiration date:
  - In the request message the old card expiration date must be provided in *encryptedOldCardInfo* object, and the new card expiration date must be filled in *encryptedNewCardInfo* object. The same PAN must be given in both objects.

API endpoint	Method
Sandbox: https://issuer-bus.test-teste-prueba.com:9215/api/v3/changecardinfo	POST
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/changecardinfo	POST

ChangeCardInfoRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Required:	Yes
Element:	**operation**
Description:	Possible values are “UPDATE”, “UNLOCK”, “SUSPEND” or “RESUME”. **Notes:** - The “**UPDATE**” operation is used by VTS, MDES and AMEX; - “UNLOCK”, “SUSPEND” and “RESUME” operations are used only **for AMEX.**
Type:	String
Required:	Yes

Element:	**operatorID**
Description:	The operator identification code.
Type:	String
Size:	0-16
Required:	Required for VTS and MDES. Not present otherwise.
Element:	**operatorName**
Description:	Operator name.
Type:	String
Size:	0-200
Required:	Required for MDES. Not present otherwise.
Element:	**operatorPhone**
Description:	Operator contact phone.
Type:	String
Size:	0-20
Required:	Required for MDES. Not present otherwise.
Element:	**reason**
Description:	The reason why a change was made.
Type:	String
Required:	Required for “VTS”and “MDES” Not present for “AMEX”
Element:	**encryptedOldCardInfo**
Description:	CardInfo - Old encrypted card information. See notes (*) for usage details.
Type:	EncryptedPayload
Required:	Yes
Element:	**encryptedNewCardInfo**
Description:	CardInfo - New encrypted card information. See notes (*) for usage details.
Type:	EncryptedPayload
Required:	Conditional

ChangeCardInfoResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Codes: “00” for Ok. “94” for Invalid Replacement PAN “95” for Cryptography Error “96” for Invalid Data “97” for Required Data Missing “99” for System Error, please check error description.

Type:	String
Required:	Yes

Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

(*) Important notes:

To update card information, it is mandatory to inform in the objects **encryptedOldCardInfo** and **encryptedNewCardInfo** the elements accordingly to the expected scenario, as indicated below:

1-) To **update PAN and ExpirationDate** is required to inform: Old PAN, Old Expiration Date, New PAN, New Expiration Date.

2-) To **update only ExpirationDate** is required to inform: Old PAN, Old Expiration Date, Old PAN, New Expiration Date.

3-) To **delete PAN** (**) is required to inform: Old PAN.

(**) Only available to issuers subscribed on Visa Account Updater (VAU).

JSON Examples

ChangeCardInfoRequest

```
{
  "requestID": "4",
  "institutionCode": "HST",
  "operation": "UPDATE",
  "reason": "About to expire",
  "operatorID": "12",
  "encryptedOldCardInfo": {
    "algorithm": "aes-ccm128",
    "nonce": "508ad7193d0b634647cdd931",
    "encryptedData":
"8ztAmsfoQdE7P22LqdAJD24VdoQay5k6mdghbKRQsPNqcNnjyl+MqDTvqqQITgolhtMawvDjnn3fO
mOJfJDvW8EeTs5ZcutGs68IKMlRGfO+xrQBFo8iXAkKEDs0qksyuj0Jm3bvWpAyXmSe4NIki4Oc+T8
plK8g/KPFHElDZVq6gJ329zmWhOMkc6GnN/Kz",
    "MACLength": 16
  },
  "encryptedNewCardInfo": {
    "algorithm": "aes-ccm128",
    "nonce": "e434a9e356425c86338c91bd",
    "encryptedData":
"H0njeQMSpIdOiuSOsILBindOGkUetIg4BoY1U+rXwf4yxeXr5f0wTru53ll6acVhZvXwqwP4xqDRG
qfQ88LN52dmt+ZfiuA2KbcPszjWkRrImg0q/tFJAuhwlKdkCcwS8+vNLrLvv56H32PB8vfJizkL0zf
/e5Y2X5jNyp7FF/D4+UHZMZzfbUA8HyQDcZ9g",
    "MACLength": 16
  }
}
```

```
Where:
//Plain OldCardInfo Object Data:
```

```
{
  "PAN": "1111110000000003",
  "expirationDate": {
    "month": "11",
    "year": "2024"
  }
}
```

```
//Plain NewCardInfo Object Data:
```

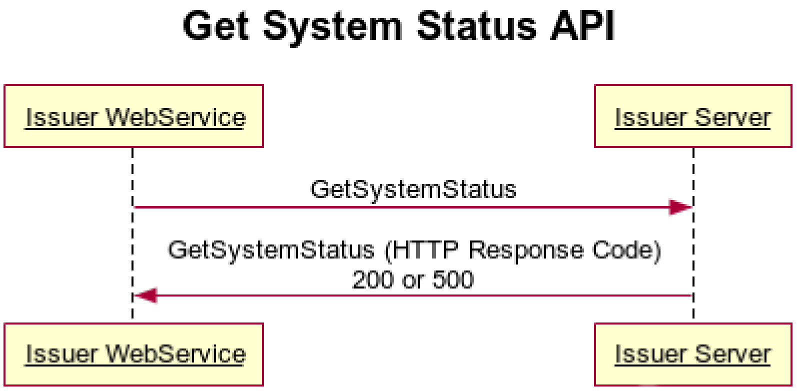
```
{
  "PAN": "1111110008484383",
  "expirationDate": {
    "month": "05",
    "year": "2026"
  }
}
```

ChangeCardInfoResponse

```
{
  "requestID": "4",
  "returnCode": "00"
}
```

5.7. GetSystemStatus

This API is used to check the system’s health status.



API endpoint	Method
Sandbox: https://issuer-bus.test-teste-prueba.com:9215/api/v3/getsystemstatus	GET
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/getsystemstatus	GET

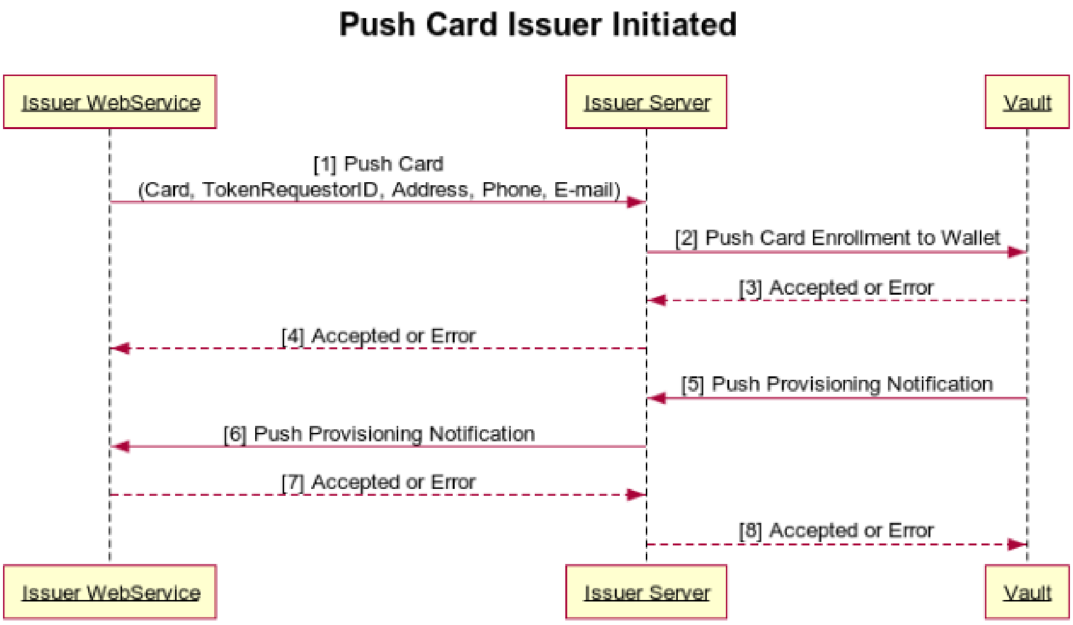
Issuer Server responds with 200 if OK or 5XX in case of error or unavailability.

5.8. PushCard

This API allows issuers to push card to wallet providers. It initiates a push provisioning, the brand will validate the request and send an acknowledgment back to the issuer. Upon successful

validation, the brand will forward the provisioning request to token requestors. One request can send push provisioning to multiple token requestors, which are associated with the same PAN and email address or phone number.

This API is only used for VISA implementations.


Push Card Issuer Initiated
Issuer WebService
Issuer Server
Vault
[1] Push Card
(Card, TokenRequestorID, Address, Phone, E-mail)
[2] Push Card Enrollment to Wallet
[3] Accepted or Error
[4] Accepted or Error
[5] Push Provisioning Notification
[6] Push Provisioning Notification
[7] Accepted or Error
[8] Accepted or Error
Issuer WebService
Issuer Server
Vault

API endpoint	Method
Sandbox: https://issuer-bus.test-teste-prueba.com:9215/api/v3/pushcard	POST
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/pushcard	POST

PushCardRequest

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**tokenRequestorID**
Description:	The token requestor identifier. It identifies Click To Pay, SamsungPay, ApplePay, a Multi Issuer Wallet, an Issuer Wallet or a Merchant. All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project. If provided, results will only contain tokens related to that specific Token Requestor ID.
Type:	String
Required:	Yes

Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes
Element:	**encryptedCardProfile**
Description:	Encrypted CardProfile. Contains of card information to be used for the payment instrument. NOTICE: The card profiles sent to Click To Pay must not provide CVV2.
Type:	EncryptedPayload
Required	Yes

PushCardResponse

Element:	**requestID**
Description:	Request identifier unique generated for each request by HST.
Type:	String
Required:	Yes
Element:	**returnCode**
Description:	Return Code: “00” for OK.
Type:	String
Required:	Yes
Element:	**errorDescription**
Description:	Error description returned only in error conditions for troubleshooting purpose.
Type:	String
Required:	Optional

JSON Examples

PushCardRequest

```
{
  "requestID": "202107270001",
  "tokenRequestorID": "40010075338",
  "institutionCode": "HST",
  "encryptedCardProfile": {
    "algorithm": "aes-gcm256",
    "iv": "2415F6220825A8BC7B7A47233F46C378",
    "encryptedData":
"GK5NfIXesgJ8loyzqKOJh4Zhg7Lbf3fzsVre43iU3F4qRv1zGTIseLteLYHUMNze1gTO186aPzMPM
lOuL4f3S3CI7b0bzOcmfxadk2hVq6/A",
    "MACLength": 12
  }
}
```

Where:

//Plain CardProfile Object Data

```
{
  "cardInfo": {
    "PAN": "4166875806119746",
    "expirationDate": {
      "month": "11",
      "year": "2024"
    },
    "cardholderName": "FRANCISCO PEREIRA"
  },
  "billingAddress": {
    "state": "CA",
    "line1": "line1",
    "line2": "line2",
    "postalCode": "94404",
    "countryCode": "US",
    "city": "FosterCity"
  },
  "provider": {
    "clientAppID": "SRC",
    "clientID": "33ba540a-20a2-2d35-4678-12502a2cde01",
    "isIDnV": false,
    "isTsAndCsAccepted": true,
    "intent": "PUSH_PROV_ONFILE",
    "walletID": "000000000000000000001235",
    "issuerAccountID": "issuerAccountID",
    "returnURIType": "WEB",
    "returnURI": "aHR0cHM6Ly93d3cuaHN0LmNvbS5ici8",
    "clientInformation": {
      "walletID": "000000000000000000001235",
      "issuerAccountID": "issuerAccountID",
      "tokenReferenceID": "tokenReferenceID",
      "source": "ISSUER",
      "firstName": "ClientFristName",
      "middleName": "ClientMiddleName",
      "lastName": "ClientLastName",
      "locale": "en_US",
      "deviceID": "...",
      "countryCode": "US",
      "contactPhone": "+447911112345",
      "contactEmail": "client@host.xyz",
    }
  }
}
```

PushCardResponse

```
{
  "requestID": "4",
  "returnCode": "00"
}
```

NOTE: In error case, the response is:

```
{
  "requestID": "4",
  "returnCode": "98",
  "errorDescription": "Invalid Request"
}
```

5.9. BulkProvision

This API allows the issuers to schedule a bulk provision of multiple card credentials to SRC participants (Click to Pay).

5.9.1. BulkProvision – Initiate Bulk Job

This API allows to submit the cards profile information to the vault and receives a push receipt ticket for this job.

Each bulk can contain **a maximum of 1000 items**, with a limit of 50 parallel bulks allowed per institution. If maximum capacity is reached, HTTP status 429 and error code '13' will be returned in *returnDetails.returnCode,* indicating: 'Too many items enqueued to be processed. Try again later'.

API endpoint	Method
Sandbox: https://homolog-api-gateway.test-teste-prueba.com:9208/api/v3/bulkprovision/initiatebulkjob	POST
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/bulkprovision/initiatebulkjob	POST

InitiateBulkJob Request

Element: **requestID**
Description: Request identifier unique generated for each request by the Issuer.
Type: String

Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes
Element:	**vaultIdentification**

Description:	Possible values are:
	“VTS” – for Visa;
	“MDES” – for Mastercard;
	“AMEX” – for Amex;
	“PL” – for Private Label.
	Used to identify the Vault in case tokenRefID does not exist in HST database.

Type:	String
Required:	Yes
Element:	**encryptedCardProfileList**
Description:	A list of encrypted CardProfile. Contains cards information to be pushed. Max 1000 items per bulk, 50 parallel bulks per issuer. NOTICE: The card profiles sent to Click To Pay must not provide CVV2.
Type:	List<EncryptedPayload> (cyphered using SIK or JWE)
Required	Yes

InitiateBulkJob Response

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**bulkPushReceiptID**
Description:	A receipt to associated to the Bulk Provision request.
Type:	String
Size:	36
Required:	Yes
Element:	**returnDetails**
Description:	The object containing the API return details.
Type:	ReturnObject
Required:	Yes

5.9.2. BulkProvision – Get Bulk Job Status

This API allows the issuers to query a bulk provision status for a given bulkPushReceiptID.

API endpoint	Method
Sandbox: https://homolog-api-gateway.test-teste-prueba.com:9208/api/v3/bulkprovision/getbulkjobstatus	POST
Production: https://issuer-bus.shieldedtransaction.com:9215/api/v3/bulkprovision/getbulkjobstatus	POST

GetBulkJob Request

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Size: 32	32
Required:	Yes
Element:	**institutionCode**
Description:	A code generated by HST that identifies the Issuer during the request.
Type:	String
Size:	32
Required:	Yes
Element:	**bulkPushReceiptID**
Description:	A receipt to associated to the Bulk Provision request.
Type:	String
Size:	36
Required:	Yes

GetBulkJob Response

Element:	**requestID**
Description:	Request identifier unique generated for each request by the Issuer.
Type:	String
Required:	Yes
Element:	**bulkJobStatus**
Description:	One of the following values: “JOB_PENDING”, “JOB_SCHEDULED”, “JOB_FINISHED”
Type:	String
Required:	Yes
Element:	**startedAt**
Description:	Timestamp informing when the job has started or is scheduled to initiate the push provision on the brand’s vault.
Type:	String (UTC formatted)
Required:	Yes
Element:	**finishedAt**
Description:	Timestamp informing when the job has terminated its execution.

Type:	String (UTC formatted)
Required:	Optional
Example:	yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
Element:	**provisionedCardIdList**
Description:	List of **cardID** associated to successful pushed cards.
Type:	Array <String>
Required:	Optional
Element:	**errorCardList**
Description:	List of **ErrorObjects** associated to failed pushed cards.
Type:	Array <ErrorObject>
Required:	Optional
Element:	**returnDetails**
Description:	The object containing the API return details.
Type:	ReturnObject
Required:	Yes

6. General Objects

6.1. CardMetaData

Element:	**foregroundColor**
Description:	Foreground color of the Digital Wallet entry for the card. (i.e. rgb(12,225,585))
Type:	String
Size:	0-32
Required:	Optional – Only available on VTS
Element:	**backgroundColor**
Description:	Background color of the Digital Wallet entry for the card. (i.e. rgb(14,245,095))
Type:	String
Size:	0-32
Required:	Optional – Only available on VTS
Element:	**labelColor**
Description:	Label color of the Digital Wallet UI entry (“space”) for the card. (i.e. rgb(06,321,769))
Type:	String
Size:	0-32
Required:	Optional – Only available on VTS
Element:	**shortDescription**
Description:	A short description of the card.
Type:	String
Size:	0-32
Required:	Optional – Only available on VTS

Element:	**longDescription**
Description:	A long description of the card.
Type:	String
Size:	0-64
Required:	Optional – Only available on VTS
Element:	**contactEmail**
Description:	Customer Service’s e-mail of the issuer bank.
Type:	String
Size:	0-64
Required:	Optional – Only available on VTS
Element:	**contactPhone**
Description:	Customer Service’s phone number of the issuer bank.
Type:	String
Size:	0-32
Required:	Optional – Only available on VTS
Element:	**contactName**
Description:	Issuer bank’s name.
Type:	String
Size:	0-64
Required:	Optional – Only available on VTS
Element:	**termsAndConditionsID**
Description:	Issuer bank terms and conditions Id configured on the Vault.
Type:	String
Size:	0-32
Required:	Optional – Only available on VTS
Element:	**cardArtID**
Description:	Issuer bank card art Id configured on the Vault.
Type:	String
Size:	0-32
Required:	Optional – Only available on VTS
~~Element:~~	~~**productId**~~
~~Description:~~	~~Unique identifier of the card product as registered on the platform.~~
~~Type:~~	~~String~~
~~Size:~~	~~0-32~~
~~Required:~~	~~Optional for “VTS” and ”MDES”~~ ~~Required for “AMEX”~~ **Deprecated – New implementations should use “*productID*” instead**
Element:	**productID**
Description:	Unique identifier of the card product as registered on the platform.
Type:	String
Size:	0-32
Required:	Optional for “VTS” and ”MDES” Required for “AMEX”

Element:	**productName**
Description:	Card product name (description).
Type:	String
Size:	0-12
Required:	Required for “AMEX”
Element:	**productType**
Description:	Card type. For example: “CREDIT”, “DEBIT”, “PREPAID”.
Type:	String
Size:	0-64
Required:	Required for “AMEX”

6.2. TokenInfo

Element:	**token**
Description:	Token Value assigned to the PAN.
Type:	Numeric
Size:	13-19
Required:	Optional
Element:	**expirationDate**
Description:	Card’s expiration date.
Type:	ExpirationDate
Required:	Optional
Element:	**state**
Description:	“ACTIVE”, “SUSPENDED”, “INACTIVE”, “CANCELED”.
Type:	String
Required:	Optional
Element:	**type**
Description:	“HCE”, “SE”,”ECOM”,”QRCODE”,”COF”.
Type:	String
Required:	Optional
Element:	**lastTokenStatusUpdatedTimeStamp**
Description:	**Format:** yyyy-MM-ddTHH:mm:ss.SSSZ The value will be in GMT.
Type:	String
Required:	Optional
Element:	**entityOfLastAction**
Description:	“TOKEN_REQUESTOR” or “ISSUER”.
Type:	String
Required:	Optional
Element:	**deviceInfo**
Description:	It will not be present for tokens that are not device bound.
Type:	Object
Size:	1

Required:	Optional

Element:	**OTPCodeIndicate**
Description:	“PRESENT”, “NOT_PRESENT” or “EXPIRED”.
Type:	String
Required:	Optional

Element:	**OTPCodeExpiration**
Description:	**Format:** YYYY-MM-DDThh:mm:ss.SSSZ The value will be in GMT.
Type:	String
Required:	Optional

Element:	**PANLastFour**
Description:	These are the last four digits of the current PAN for the token.
Type:	String
Size:	4
Required:	Optional

Element:	**previousPANLastFour**
Description:	These are the last four digits of the previous PAN for the token. If a card has been replaced while the token was in an active state then this represent the previous PAN that the token was associated with.
Type:	String
Size:	4
Required:	Optional

Element:	**tokenRefID**
Description:	Identifier of the Token.
Type:	String
Size:	64
Required:	Yes

Element:	**activationFlow**
Description:	Defines how the token was activated. Possible values: “GREEN” or “YELLOW”.
Type:	String
Size:	64
Required:	Optional

Element:	**panSource**
Description:	Indicates how the PAN was provided. Possible values are: “**ON_FILE**” – PAN origin is a card number stored in a merchant; “**MANUALLY**” – PAN was entered by the customer; “**MOBILE_APP**” – PAN provided by a mobile app. Typically a list of cards provided by the issuer after cardholder authentication; “**TOKEN**” – The source of pan of this token (ECOM o COF) provisioning was issued by a token device bound (NFC/SE). Applicable to a scenario

	such as a wallet has a NFC/SE token and it is provisioning a new E-
Type:	Commerce/COF token.
Size:	String
Required:	64
	Optional
Element:	**activationMethod**
Description:	Describes how the token was activated: Possible values are: “AUTOMATIC” (green flow), “STEPUP_OTP”, “STEPUP_CALL_CENTER”, “STEPUP_ISSUER_APP”, “UNKNOWN”.
Type:	String
Size:	64
Required:	Optional
Element:	**activationDateTime**
Description:	GMT Date and time of activation (“yyyy-MM-ddTHH:mm:ss.SSSZ”)
Type:	String
Size:	64
Required:	Optional
Element:	**tokenAssuranceLevel**
Description:	The assurance level assigned to the token.
Type:	String
Size:	2
Required:	Optional
Element:	**tokenRequestorID**
Description:	Identification of the Token Requestor requesting digitization. It identifies SamsungPay, ApplePay, a Multi Issuer Wallet or an Issuer Wallet. All the Token Requestor ID values are generated by the brand and a table is provided by them to Issuers during the initial steps of the project.
Type:	If provided, results will only contain tokens related to that specific
Required:	Token Requestor ID.
	String
	Optional
Element:	**tokenRequestorName**
Description:	Identification of the Token Requestor Name dynamically reported by the vault, it is present in DigitizationNotificationAPI payload. When provided updates the audit reports in Pay Admin.
Type:	String
Required:	Optional
Element:	**deletedFromApp**
Description:	Indicates if the token is deleted only from the device/token requestor or both device and the MDES platform. For Apple Pay tokens deleted from the device doesn’t produce automatic
Type:	notifications to the issuers.
Required:	Boolean

	Optional – MDES Only (Apple Pay)

6.3. DeviceInfo

Element:	**deviceType**
Description:	"UNKNOWN", "MOBILE_PHONE", "TABLET", "WATCH", "PC".
Type:	String
Required:	Optional
Element:	**deviceNumber**
Description:	Mobile phone number or last four digits of mobile phone number.
Type:	String
Size:	0-13
Required:	Optional
Element:	**deviceName**
Description:	User assigned device name.
Type:	String
Size:	0-16
Required:	Optional
Element:	**deviceModel**
Description:	Model of the device.
Type:	String
Size:	0-32
Required:	Optional
Element:	**serialNumber**
Description:	Masked Serial Number.
Type:	String
Size:	0-32
Required:	Optional
Element:	**deviceID**
Description:	The unique device identifier.
Type:	String
Size:	48
Required:	Optional
Element:	**deviceIndex**
Description:	The index number from Vault where deviceID is stored. Required for token device binding.
Type:	String
Size:	2
Required:	Optional

6.4. AuthenticationMethod

Element:	**identifier**
Description:	Required if cardholder verification method is returned. Identifies each verification method during the issuer response, which means is **unique** and opaque identifier for each method. This ID should be defined and provided by the issuer.
Type:	String
Size:	0-32
Required:	Yes
Element:	**type**
Description:	The available options are: “**cell_phone**” – OTP sent to cell phone number; “**email**” – OTP sent to e-mail address; “**bank_app**” – Authentication through the issuer app; “**customer_service**” – Authentication through issuer call center; “**outbound_call**“ – Call received by the cardholder.
Type:	String
Required:	Yes
Element:	**maskedInfo**
Description:	Masked Consumer (cell phone): ‘********19’ Masked Consumer (email_address): ‘ip****@gmail.com’. Mobile Banking (bank_app): ‘Mobile Banking App’ Call Center (customer_service): ‘1-800-555-555’
Type:	String
Size:	0-64
Required:	Yes
Element:	**customerAddress**
Description:	email: ‘testcustomer@gmail.com’. phone number: ‘1-800-555-555’
Type:	String
Size:	0-64
Required:	Optional – only for auditing purpose
Element:	**sourceAddress**
Description:	When used with ‘Type’ ‘bank_app’, this value must contain the appropriate identifier for the associated issuer mobile banking application, such as *“com.DemoBank.DemoApp”* for example. For Apple this would be the Apple Adam ID and for Android this would be
Type:	the Android Package name.
Size:	String
Required:	0-64
	Optional – only used for Bank App flow
Element:	**platform**
Description:	This field is used when the **Type** field contains the value **bank_app.** Valid Values: “IOS”, “ANDROID”, “WINDOWS”, “WEB”.
Type:	String
Required:	Optional

6.5. ExpirationDate

Element:	**month**
Description:	Month of expiry date.
Type:	String
Size:	2
Required:	Yes
Element:	**year**
Description:	Year of expiry date (i.e. **XXXX**).
Type:	String
Size:	4
Required:	Yes

6.6. CardInfo

Element:	**PAN**
Description:	Primary Account Value.
Type:	String
Size:	16-19
Required:	Yes
Element:	**expirationDate**
Description:	Card expiration date.
Type:	ExpirationDate
Required:	Optional - Required for Click to Pay
Element:	**CVV2**
Description:	Card Verification Value presented on the back of the physical card.
Type:	String
Size:	3
Required:	Optional
Element:	**cardholderName**
Description:	Cardholder Name as it appears on card. Special characters or numbers are not valid.
Type:	String
Size:	Max 32
Required:	Optional - Required for Click to Pay
Element:	**PANSequence**
Description:	Funding account PAN sequence. Examples: 00 (Default Value), 01, 02, 03.
Type:	String
Size:	Up to 3
Required:	Required for “AMEX”. Conditional for MDES – Present when *Tap to Add Card* feature is enabled and if chipdata is present in the encryption payload

6.7. EncryptedPayload

Element:	**algorithm**
Description:	Encryption Algorithm used to protect data. Supported types are: “**aes-gcm128**”, “**aes-ccm128**”, “**aes-gcm256**”, “**aes-ccm256”,”jwe”.** Refer to https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf and https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf
Type:	String
Required:	Yes
Element:	**nonce**
Description:	Nonce for AES_CCM.
Type:	String
Size:	7, 8, 9, 10, 11, 12 or 13 (if not sure, 11 should be used)
Required:	Optional
Element:	**iv**
Description:	Initial Vector for AES_GCM. This field is a String which represents an
Type:	array of 32 hexadecimal digits, representing at most 16 bytes.
Size:	String
Required:	32
	Optional
Element:	**encryptedData**
Description:	Encrypt Data value using SIK. **All the ciphered data must be transmitted in base64.**
Type:	String
Size:	0-256k
Required:	Yes
Element:	**associatedData**
Description:	Data that is not encrypted but used for MAC calculation.
Type:	String
Size:	0-256K
Required:	Optional
Element:	**MACLength**
Description:	Specifies the MAC length that will be generated. The MAC contents are located at the end of the encryptedData element. Valid values for **CCM algorithm**: 4, 6, 8, 10, 12, 14 or 16 bytes (reasonable minimum is 12). Valid values for **GCM algorithm**: 4, 6, 8, 10, 12, 14 or 16 bytes (reasonable minimum is 12).
Type:	Numeric
Required:	Yes

6.8. TokenUserInfo

Element:	**ID**
Description:	The unique value that identifies the token user. (The entity which initiates the payment request).
Type:	String
Size:	11
Required:	Yes
Element:	**appType**
Description:	Application type for the token user. This entity can be the merchant, a marketplace, or a checkout host. Possible values are: “WEB” “MOBILE_APP” “MOBILE_WEB” “MARKETPLACE” “VOICE_APP” “BIOMETRIC_APP”
Type:	String
Required:	Optional

6.9. MerchantInfo

Element:	**ID**
Description:	The unique value that identifies the merchant. Required for trusted listing enrollment.
Type:	String
Size:	8
Required:	Optional
Element:	**merchantName**
Description:	The merchant name.
Type:	String
Size:	256
Required:	Optional

6.10. RiskInformation

Element:	**recommendedDecision**
Description:	The decision recommended by the Wallet Provider (token requestor). Possible values are: “GREEN”, “YELLOW”, “ORANGE” or
Type:	“RED”.
Size:	String
Required:	64

	Optional
Element:	**deviceScore**
Description:	Score given to the device by the Wallet Provider (token requestor). Value between 1 and 5, where 5 indicates the most confidence on the device.
Type:	String
Size:	2
Required:	Optional
Element:	**accountScore**
Description:	Score given to the account by the Wallet Provider (token requestor). Value between 1 and 5, where 5 indicates the most confidence on the account.
Type:	String
Size:	2
Required:	Optional
Element:	**vaultRiskAssessmentScore**
Description:	Advanced Authorization (AA) Risk Score is generated by VisaNet on the last payment transaction for the PAN. Values are 00 through 99, with a higher value indicating higher risk. This value is calculated by the AA Risk engine, based on the PAN’s transaction pattern (long-term and short-term).
Type:	String
Size:	2
Required:	Optional for VTS. Not present otherwise.
Element:	**vaultTokenScore**
Description:	Value indicating the degree of risk associated with the token. Numeric value is 00 through 99. A score of 00 indicates that card brand did not provide a score.
Type:	String
Size:	2
Required:	Optional for VTS. Not present otherwise.

6.11. TermsAndConditions

Element:	**id**
Description:	The terms and conditions identifier generated by the Vault.
Type:	String
Size:	64
Required:	Optional
Element:	**date**
Description:	The date and time the terms and conditions were accepted by the cardholder. **Format:** YYYY-MM-DDThh:mm:ss.SSSZ The value will be in GMT.

Type:	String
Size:	0-32
Required:	Optional

6.12. Market

Element:	**countryCode**
Description:	Two letter country code based on ISO 3166. Example: “BR”, “US”, “MX”.
Type:	String
Size:	2
Required:	Required for “AMEX”.
Element:	**regionName**
Description:	Region name is the country name.
Type:	String
Size:	0-64
Required:	Required for “AMEX”.
Element:	**locale**
Description:	Locale in xx_XX format. The format is based on xx_XX, where xx refers to Language code and XX refers to Country code. Examples: en_US, en_SA, pt_BR, es_MX, etc. Note: ISO standard values for the country of the Issuer.
Type:	String
Size:	0-12
Required:	Required for “AMEX”.

6.13. CardProfile

Element:	**cardID**
Description:	Card identification provided by the issuer. This will be used as return to the issuer when receiving the bulk provision results.
Type:	String
Size:	0-36
Required:	**Conditional – Mandatory for bulk operations**.
Element:	**cardInfo**
Description:	Card information.
Type:	CardInfo
Required:	Yes
Element:	**billingAddress**
Description:	Billing Address associated with the payment instrument.
Type:	BillingAddress
Required:	Yes
Element:	**provider**
Description:	

	Information about the provider of the payment instrument and the contexto under which it is provided.
Type:	Provider
Required:	Yes

6.14. BillingAddress

Element:	**line1**
Description:	First line associated with the address.
Type:	String
Size:	64
Required:	Optional
Element:	**line2**
Description:	Second line associated with the address.
Type:	String
Size:	64
Required:	Optional
Element:	**city**
Description:	City associated with the address.
Type:	String
Size:	32
Required:	Optional
Element:	**state**
Description:	State or province code associated with the address.
Type:	String
Size:	64
Required:	Optional
Element:	**postalCode**
Description:	The postal code associated with the address.
Type:	String
Size:	10
Required:	Optional
Element:	**countryCode**
Description:	For VTS: Two letters country code based on ISO 3166-1 alpha 2. Example: “BR”, “US”, “MX”. For MDES: Three letters country code based on ISO 3166-1 alpha 3. Example: “BRA”, “USA”, “MEX”.
Type:	String
Size:	2-3
Required:	Required

6.15. Provider

Element:	**intent**

Description:	The intent of the encryption; what is the encryption of the data trying to do. For VTS Secure Remote Commerce, specify PUSH_ PROV_ONFILE.
Required:	Required
Element:	**walletID**
Description:	Identifier of the wallet that generated the request. Some wallet providers such Apple Pay and Google pay define it with the same value used for device identification. For Apple Pay, this field shows the SEID.
Type:	String
Size:	64
Required:	Optional
Example:	Format at HST WhiteLabel Wallet: N3GN-KWH6-NTYC-QNKN
Element:	**clientID**
Description:	Unique ID that identifies the SRC entity on the vault. VTS ClickToPay on VCEH: “33ba540a-20a2-2d35-4678-12502a2cde01”
Type:	String
Size:	36
Required:	Required
Element:	**clientAppID**
Description:	Unique Identifier for the client application, used to provide some of the encrypted values. Example: Issuer’s AppID (vClientAppID) used to select the PAN and the wallet.
Type:	String
Size:	36
Required:	Optional
Element:	**isIDnV**
Description:	Whether the issuer wants ID&V to be performed. The value is “true” or “false”.
Required:	Optional
Element:	**isTsAndCsAccepted**
Description:	Use to indicate to the wallet provider whether or not the customer already accepted the issuer terms and conditions up-front. Supported values are: “true” or “false”. The Visa Click to Pay SRC platform requires “true”.
Required:	Required
Element:	**issuerAccountID**
Description:	Uniquely represents “pushing” account from issuer system. May be different from PAN holder account.
Type:	String
Size:	24
Required:	Yes
Element:	**clientInformation**
Description:	Client’s information.
Type:	ClientInformation
Required:	Optional
Element:	**returnURIType**

Description: The kind of URI for the return app.
Format: It is one of the following values:
• IOS— iOS app
• ANDROID— Android app
• WEB— Browser-based app
Required: Optional

Element: **returnURI**
Description: URI provided by the issuer to the token requestor to return control to the issuer app. This can be an app or a web URL.
Type: String
Size: 512
Required: Optional

6.16. ClientInformation

Element: **source**
Description: Indicates the source of the information
The value can be “ISSUER” or “TOKEN_REQUESTOR” .
Required: Optional

Element: **walletID**
Description: Identifier of the wallet that generated the request. Some wallet providers such Apple Pay and Google pay define it with the same value used for device identification. For Apple Pay, this field shows the SEID.
Type: String
Size: 64
Required: Optional
Example: Format at HST WhiteLabel Wallet: N3GN-KWH6-NTYC-QNKN

Element: **firstName**
Description: First name of client. Issuer to populate with the information they have for the client.
Type: String
Size: 80
Required: Required

Element: **middleName**
Description: Middle name of the client. Issuer to populate with the information they have for the client.
Type: String
Size: 80
Required: Required

Element: **lastName**
Description: Last name of the client. Issuer to populate with the information they have for the client.
Type: String
Size: 80

Required:	Required
Element:	**issuerAccountID**
Description:	Issuer account ID as provided by the issuer to the token requestor.
Type:	String
Size:	24
Required:	Required
Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PA NRefID for furt her use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others.
Type:	String
Size:	64
Required:	Optional
Element:	**contactPhone**
Description:	Mobile phone number of the client as per issuer records. The format follows E.164 standard – Example: +44791112345.
Type:	String
Size:	0-32
Required:	Required
Element:	**contactEmail**
Description:	Email address of client as per issuer records.
Type:	String
Size:	0-64
Required:	Required
Element:	**countryCode**
Description:	Two letters country code based on ISO 3166. Example: “BR”, “US”, “MX”.
Type:	String
Size:	2
Required:	Optional
Element:	**locale**
Description:	Locale in xx_XX format. The format is based on xx_XX, where xx refers to Language code and XX refers to Country code. Examples: en_US, en_SA, pt_BR, es_MX, etc. Note: ISO standard values for the country of the Issuer.
Type:	String
Size:	0 – 12
Required:	Yes
Element:	**deviceID**
Description:	The unique device identifier.
Type:	String

Size:	24
Required:	Optional
Element:	**externalConsumerID**
Description:	Unique identifier of a consumer provided by the Issuer
Type:	String
Size:	100
Required:	Conditional / Required for Click to Pay (Visa only)

6.17. PushNotification

Element:	**source**
Description:	Indicates the source of the information The value can be “ISSUER” or “TOKEN_REQUESTOR” .
Required:	Required
Element:	**firstName**
Description:	First name of client. Issuer to populate with the information they have for the client.
Type:	String
Size:	80
Required:	Required
Element:	**middleName**
Description:	Middle name of the client. Issuer to populate with the information they have for the client.
Type:	String
Size:	80
Required:	Required
Element:	**lastName**
Description:	Last name of the client. Issuer to populate with the information they have for the client.
Type:	String
Size:	80
Required:	Required
Element:	**contactPhone**
Description:	Mobile phone number of the client as per issuer records. Phone numbers do not contain country codes.
Type:	String
Size:	0-32
Required:	Required
Element:	**contactEmail**
Description:	Email address of client as per issuer records.
Type:	String

Size:	0-64
Required:	Required
Element:	**locale**
Description:	Locale in xx_XX format. The format is based on xx_XX, where xx refers to Language code and XX refers to Country code. Examples: en_US, en_SA, pt_BR, es_MX, etc. Note: ISO standard values for the country of the Issuer.
Type:	String
Size:	0 – 12
Required:	Required
Element:	**tokenRefID**
Description:	Token Reference ID associated to the token created to the specified card (EncryptedCardInfo) on the specified device (WalletID/DeviceID). It is a value assigned by the vault. Each token generates a tokenRefID value, which means a PAN can have one or more tokenRefID values assigned. By using this data it is not necessary to input the real token value. It is recommended to associate or to bind the tokenRefID value with the PA NRefID for furt her use in APIs such as Get TokenInfo, ChangeTokenStatus, ActivateToken and others.
Type:	String
Size:	64
Required:	Required
Element:	**deviceID**
Description:	The unique device identifier.
Type:	String
Size:	24
Required:	Required

6.18. ReturnObject

Element:	**returnCode**
Description:	The error code associated to the API business error. “00” – OK.
Type:	String
Size:	16
Required:	Required
Element:	**errorDetails**
Description:	The error details associated to the API business error.
Type:	ErrorObject
Required:	Optional

6.19. ErrorObject

Element:	**cardID**

Description:	Card identification provided by the issuer. This will be used as return to the issuer when receiving the bulk provision results.
Type:	String
Size	0-36
Required:	**Conditional – This field is required in the GetBulkJob request, specifically in the errorCardList element**.

Element:	**errorCode**
Description:	The error code associated to the API business error.
Type:	String
Size:	16
Required:	Required

Element:	**errorDescription**
Description:	The error description associated to the API business error.
Type:	String
Size:	256
Required:	Required

6.20. ChipData

Element:	**iccSystemRelatedData**
Description:	The Account Primary Account Number of the card to be digitized. Contains Hex-encoded string.
Type:	String
Size:	Up to 255

Element:	**track2Data**
Description:	The track2Data of the card to be digitized. Contains Hex-encoded string.
Type:	String
Size:	Up to 37

Element:	**panEntryMode**
Description:	The method used for PAN entry to initiate the digitization. One of the possible values: “05” = Automatic entry of the PAN into the merchant terminal through an integrated circuit card (reserved for the DIP Card). “07” = Automatic entry of the PAN into the merchant terminal via Contactless M/Chip.
Type:	String
Size:	Up to 4

6.21. ChipDataValidationResult

Element:	**OBSServiceIndicator**
Description:	

	The On-behalf Service indicator of the chip Data validation. One of the possible values: “02” = Pre-validation Service of the M/Chip Cryptogram
Type:	“03” = Validation of the M/Chip Cryptogram in Stand-In Processing
Size:	String
	Up to 2
Element:	**OBSResult**
Description:	The On-behalf Service chip Data validation result. One of the possible values is valid: “A” = Valid Application Cryptogram (AC); ATC out of permitted range “F” = Format Error “K” = Key file does not match for this combination of PAN, PAN expiration date and KDI “U” = Unable to process “V” = Valid “X” = Security platform interruption “Z” = Security platform processing error
Type:	String
Size:	Up to 3

7. Return Codes

Code	**Description**
00	Ok
05	Card not eligible
11	Invalid Institution Code
13	Too many items enqueued to be processed, try again later
16	Card not found, invalid PAN
22	Invalid Card Security Code
23	Invalid Card Expiration Date
24	Card has not been activated, replaced, or renewed card has not been activated
25	Non-whitelisted accounts when a market is at beta test phase
26	Ineligible instant account/instant membership account provisioning
27	Too many attempts, suspected fraud. Return expected when element "recommendedDecisionReasonCode" value received in Check Eligibility request is "0002"
85	Requires ID&V

90	Connection Timeout Error
91	Invalid Token Status/Token Not Active
92	Token Not Found
93	Token Already in the State Requested
94	Invalid Replacement PAN
95	Cryptography Error
96	Invalid Data
97	Required Data Missing
98	Invalid Request
99	System Error

8. Revision History

Date	Version	Description	Author
06/05/2019	3.0	This version of the document received updates to contain information about the I-TSP only.	Victor Nascimento, Alexandre Rosa, José Antonio Ramos, Adriano Domingues
07/04/2019	3.1	- The institutionCode element was included on the request of all APIs.	Alexandre Rosa
03/23/2020	3.2	- The DeviceBindingEligibility API was included; - TokenUserInfo and MerchantInfo Objects were created; - The LifeCycleNotification API received updates: - The event element received new types: “DEVICE_BINDING” and “TRUSTED_LISTING”; - tokenUserInfo, merchantInfo, deviceBindingResult and trustedListingResult elements were created. - The element otpReasonCode was included on the SendPassCode API; - The element deviceBindingInfo was included on the GetTokenInfo API; - The DeviceInfo Object received new elements: deviceID, deviceName and deviceIndex - The ChangeTokenStatus API received updates: - The element deviceInfo was included; - The element merchantInfo was included; - The element action received new types: “DEVICE_BINDING_APPROVE”, “DEVICE_BINDING_REMOVE”, “TRUSTED_LISTING_ADD”, “TRUSTED_LISTING_REMOVE” - The element source of the CheckEligibility API received a new type: “TOKEN”	Alexandre Rosa, José A. Ramos, Victor Nascimento

		- The algorithm element from EncryptedPayload Object description was updated to inform that the "none" algorithm is used only for testing. - The CheckEligibility API description received updates. - The requestID, processID, institutionCode, tokenRequestorID, tokenRefID, PANRefID, errorDescription, encryptedCardMetaData, authenticationMethods, userID and messageDetail element descriptions received updates to inform more details. - The ChangeTokenStatus API description received updates. - The ChangeCardInfo API description received updates. - The ActivateToken API description received updates. - The GetPANByPANRefID API received updates. - The vaultIdentification element was included on GetTokenInfo, ChangeTokenStatus and ActivateToken APIs - The userLanguage and PANRefID elements were changed from "Required" to "Optional – required for VTS only" - The event and actionResult elements from DigitizationNotification API received updates to inform new possible values - The standInReasonCode element was included on the request of DigitizatioNotification API	
07/14/2020	3.3	- New RiskInfo object was added to be used as an optional element in *CheckEligibility* API, in order to support the Issuers in the decision for card digitization eligibility, based on the information received by the Wallet provider. - Event INACTIVE added in LifeCycleNotification API.	José A. Ramos, Eduardo Cunha

		- New optional elements were added in TokenInfo Object - activationFlow, panSource, activationMethod and activationDateTime. - Added new Outbound API: changeCardInfoNotification. - Change Card Info API description was changed, to include MDES constraints about account range.	
04/01/2021	3.4	- The tokenType, tokenRequestorName, recommendedDecision and recommendedDecisionReasonCode new elements were included on the request of CheckEligibility API - The “22” and “23” new returnCodes were included on the response of CheckEligibility API - The encryptedCardMetaData element was deprecated - The cardMetaData element was included on the response of CheckEligibility API. Also, new information about the Card Meta Data Implementation Options for Issuers was included - The encryptedCardInfo element was included on the request of SendPassCode API - The “PENDING_ACTIVATION”, “NO_FIRST_PURCHASE” and “NO_RECENT_PURCHASE” values of the Event element were included on the request of LifeCycleNotification - The requestID element description was changed on all the request of the Inbound APIs and response of the Outbound APIs - The tokenRequestorID element was included on the request of GetTokenInfo, ChangeTokenStatus and ActivateToken APIs - The 22 (Invalid Card Security Code) and 23 (Invalid Card Expiration Date) new Return Codes were included in section 7	Eduardo Cunha, Alexandre Rosa, José Ramos, Victor Nascimento

- The deviceInfo element was included on the request of the LifeCycleNotification API

- The termsAndConditions element was included on the request of the DigitizationNotification API

- The TermsAndConditions object was created

- The values from the element deviceBindingResult of LifeCycleNotification API were changed

- The status element title was changed to state on TokenInfo object

- The PANRefID element description on all APIs was updated to inform the differences between VISA and Mastercard scenarios

- The tokenInfo element was included on the request of the DigitizationNotification API

- The activationCode element on the request of the ActivateToken API was deprecated and will no longer be used

- The elements description of the AuthenticationMethod object received updates to detail the cases

- The operatorName and operatorID elements description on the request of the ChangeCardInfo API received updates

- The processID element was included on the request of the LifeCycleNotification API. The description of this element on the request of the CheckEligibility API received updates

- The encryptedCardInfo element on the request of the DigitizationNotification API was changed from ‘optional’ to ‘required’

- The deviceBindingInfo element on the request of the GetTokenInfo API was changed from ‘required’ to ‘optional’

- The Backward Compatibility session (1.1) description was updated to ensure the details to the issuers

		- The deviceBindingResult element on the request of the LifeCycleNotification API received a new value “DEVICE_BINDING_REMOVED”	
08/09/2021	3.5	- The cardMetaData element description was adjusted to described that this element is not encrypted on the response of CheckEligibility. - JSON Example of CheckEligibility Request was updated. - A note about MDES notifications was included on the description of the DigitizationNotification API. - The “token” element name was fixed on the DigitizationNotification Request. The previous version of the document incorrectly showed this element as “tokenInfo” in API description and example. - The example was fixed to not display a list but display as a single object for “riskinfo” in Check Eligibility API. - The OTPExpiration element format was fixed on the SendPassCode Request. - Authentication method “email_address” was fixed on SendPassCode and AuthenticationMethod object from “email_address” to “email”. - Size parameter was included on “processID” element in all APIs that have this field. - The “tokenRefId”, “PANRefID” and “processID” elements description on the request of the CheckEligibility API received updates - The “tokenRefId” and “PANRefID” elements description on the request of the DigitizationNotification API received updates - The “tokenRefId” and “processID” elements description on the request of the LifeCycleNotification API received updates.	Jose Antonio Ramos, Rafaela Laurencini
08/17/2021	3.5.1	- Major changes were performed to support integration with AMEX brand, considering the APIs’s elements, values, and objects.	José Antonio Ramos, Rafaela Laurencini, Gabriel Brogni Zaccaron

		- The new object Market were included – used only for AMEX. - The "24", "25" and "26" new returnCodes were included on the response of CheckEligibility API, also in the list of the "Return Codes" – used only for AMEX. - The elements "market", "expirationDate" and "PANSequence" were included on the response of CheckEligibility API – used only for AMEX. - The element "PANSequence" was included on the CardInfo Object – used only for AMEX. - The "vaultIdentification"and "cardkey" elements were included on the request of the GetAssociatedTokens API – used only for AMEX. - UNLOCK, SUSPEND and RESUME values were included in the "operation" element on the ChangeCardInfo API request – used only for AMEX. - The "DEVICE_BINDING" value was fixed to "DEVICE_BINDING_RESULT" in the "event" element on the LifeCycleNotification API – used only for VTS.	
09/15/2021	3.6	- The "PushProvisioningNotification" and "PushCard" API were included. - The "CardProfile", "BillingAddress", "Provider" and "ClientInformation", "PushNotification" new objects were created. - The "source" element description on the request of the CheckEligibility and DigitizationNotification APIs received update. - The "panSource" element description on the TokenInfo object received update. - JSON Example of CheckEligibility Responses was updated.	Rafaela Laurencini, Danilo Santana e Silva, José Antonio Ramos, Victor Nascimento.

12/01/2022	3.7 (22.12)	- Added “0004” as a new code for recommendedDecisionReasonCode in CheckEligibility API (Request). - Added new returnCode “16” and “27” in Check Eligibility API (Response). - General revision of the Return Codes section, removing some the error codes that do not apply to Issuer Server APIs. - Updated return codes list in the following APIs: ChangeTokenStatus, ChangeCardInfo and GetTokenInfo – adding more specific errors. - Inbound APIs endpoints were adjusted. - Optional tokenAssuranceLevel element was included in TokenInfo Object. - Optional DeviceID information added in DigitizationNotification API. - Updated GetTokenInfo API setting field tokenRefID as required, also for MDES. - ChangeCardInfoNotification API - updated encrypted field in sample request, matching the case of PAN field. - TokenRequestorID included as optional field in GetTokenInfo API. - Included tokenInfoList a new field in GetAssociatedTokens API. - Review of fields “operatorID”, “operatorName”and “operatorPhone” in the following APIs: “GetAssociatedTokens”, “GetTokenInfo”, “ChangeTokenInfo”, “ChangeTokenStatus”, “ActivateToken” and “ChangeCardInfo”. - Updated field dateTime on DigitizationNotification and Token Life Cycle From: YYYY-MM-DDThh:mm:ss.SSSZ To: YYYY-MM-DDThh:mm:ss.SSS.	Danilo Santana e Silva, José Antonio Ramos.

		- Added new fields to LifecycleNotification Request, including encrypted TOKEN and PAN when informed by the vault. - Added new optional field into Token Info Object: tokenRequestorName. - Updated “IV” description in EncryptedPayload. - LifeCycleNotification API - process ID field is now Optional due to Visa Cloud Token Framework – for CTF Flows, this element is not sent by VTS. - LifeCycleNotification API – Two new events included for MDES “DELETED_FROM_CONSUMER_APP” and “REDIGITIZATION_COMPLETE”. -TokenRequestorName is now optionally available in the outbounds CheckEligibility and DigitizationNotification APIs requests, the former presents this field in parent level while the latter is encapsulated in TokenInfo Object. - Optional field deviceModel was inserted in DeviceInfo object. - GetTokenInfo requires tokenRequestorID to be informed in the issuer request (only VTS), due to VTS basic tokenization. - Element tokenRefID now is also present for MDES in Check Eligibility API Request. - Onboarding environment section updated, added testing SIK components.	
02/28/2023	3.7.1 (23.02)	- Added new value “BROWSER” for source field on CheckEligibility Request. - Supress of duplicated field “tokenRequestorName” on CheckEligibility Request.	Danilo Santana e Silva

		- Field “tokenRequestorID” on PushCard Request is now mandatory.	
08/15/2023	3.8 (23.08)	- Updated walletID field description, for Apple Pay it represents the SEID. - New BulkProvision API implemented at section 5.9; new BulkProvisionNotification API implemented at section 4.9; new General Objects ReturnObject and ErrorObject at section 6.18 and 6.19 - Added new riskRecommendedDecision “0005” in Checkeligibility API	Danilo Santana e Silva
02/28/2024	3.8.1 (24.02)	- The *riskInformation* at HST CheckEligibility API request, isn’t TR exclusive anymore, it may be reported also by the brand’s vault. - New optional field *vaultTokenScore* included in RiskInfo object. - New optional field *tokenInfo* included in DeviceBindingEligibility API. - Added several new *recommendedDecisionReasonCode* in CheckEligibility API - Card Meta Data object field updated from *productId* to *productID*, the previous format is kept for compatibility. - The VTS restrictions were updated for ChangeCardInfo API. - Billing Address object requires countryCode field on push to ClickToPay. - Added comments on ChangeTokenStatusAPI and CheckEligibilty API for VTS reprovision of HCE deleted tokens that occur in 7 days. - Added *recommendedDecisionReasonCodeList* at the CheckEligibility API request. - Added new return code (90) to represent *connection* that are suspended due to timeouts. - Enhanced *walletID* field description.	Danilo Santana e Silva, José Antonio Ramos

		- New optional field, “deletedFromApp”, was included in TokenInfo object. - Removed CVV from Push Card API code sample. This field is not present for ClickToPay. - CheckEligibilityReasonCode “0005”, supports YELLOW flow recommended decision mapping when required by the issuer.	
05/15/2024	3.8.2 (24.05)	- Element *institutionCode* was added in outbound PushProvisioningNotification and BulkProvisionNotification API requests (Push card to Click to Pay Use cases).	Daniel P. Santo, José A. Ramos
06/28/2024	3.8.3 (24.06)	- Add a new value “***CONTACTLESS_TAP***” in Element *Source* at the CheckEligibility API request - Add a new value “***PC***” in Element *DeviceType* at section *6.3 Device Info.* - Removed duplicated value “***TABLET***” in Element *DeviceType* at section *6.3 Device Info.* - Adjusted the name of element **isTsAndCSAccepted** to **isTsAndCsAccepted** at section 6.15 (*Provider Object*). - Elements *tokenRefID* and *deviceID* on Section *6.16 ClientInformation* are now optional. - Changed the size of element *bulkPushReceiptID* on the *GetBulkJobStatus* API Request to 36.	Daniel P. Santo
10/30/2024	3.8.4 (24.10)	- Adjusted broken hyperlinks. - New elements *chipData* and *chipDataValidationResult* were included in Check Eligibility request. - New Object 6.20 *ChipData* was added in General Objects. - New Object 6.21 *ChipDataValidationResult* was added in General Objects. - Element *PANSequence* in *CardInfo* object was changed.	Daniel P. Santo, José A. Ramos

		- New element *chipDataValidationResponse* was added in Check Eligibility Response - Element Source in DigitizationNotification request API was changed to support a new value (“CONTACTLESS_TAP”). - In ActivateToken API request, tokenRequestorID field is now required.	
12/12/2024	3.8.5 (24.12) Alpha	Description of element *processID* was changed in the request of the following APIs: Check Eligibility, Digitization Notification, SendPasscode and Lifecycle Notification.	José Antonio Ramos
01/22/2025	3.8.6 (25.01)	Element *externalConsumerID* was included in the field ClientInformation, used in PushCard and InitiateBulkJob APIs	Mariana R. F. Sampaio
05/16/2024	3.8.7	- Added PAN replacement instructions for Visa cards to the ChangeCardInfo API; - Added item and institution limits to the InitiateBulkJob API request; - Corrected the object name *riskInformation* in the CheckElegibility API request and the GetTokenInfo API response; - Added a size attribute to the *cardID* element in the *cardProfile* object of the PushCard API request; - Added a format to the *contactPhone* element in the *ClientInformation* object, used in the following APIs request: PushProvisioningNotification, PushCard and InitiateBulkJob; - The *cardHolderName* element in the *CardInfo* object is now mandatory for Click to Pay cases in the InitiateBulkJob API request. - Added support for the JWE algorithm in the *EncryptedPayload* object, used in InitiateBulkJob API request.	Mariana R. F. Sampaio

		- A new return code '13' has been added to the InitiateBulkJob API response and documented in the 'Return Codes' list. - The *operation* field in the ChangeCardInfof API request was updated: the DELETE operation is no longer available.	
07/04/2025	3.8.8	- The *locale* element in the *ClientInformation* object is now required in the PushCard API request; - Changed the Sandbox URLs for the InitiateBulkJob and GetBulkJobStatus APIs; - The *expirationDate* element is now mandatory in the 'CardInfo' object for Click to Pay requests in the InitiateBulkJob API; - Added a size attribute for the *bulkPushReceiptID* element in the BulkProvisionNotification API request and InitiateBulkJob API response. - Added a new optional element in Check Eligibility Request – "deviceType". - Added a new optional element in Digitization Notification Request API – "tokenRequestorName".	Mariana R. F. Sampaio