Remote Messaging

This section details the messaging requirements for connecting between ActiveAccess and an issuer's remote systems.

The ActiveAccess authentication system is responsible for managing the authentication of cardholders during American Express SafeKey, Diners Club International ProtectBuy, JCB J/Secure, Mastercard SecureCode / Identity Check and Verified by Visa / Visa Secure transactions. In order to support this requirement, the system must have access to appropriate information in order to uniquely determine the identity of the cardholder, whether a cardholder transaction requires authentication and what type of authentication is required. The determination of whether a cardholder is registered, whether a transaction requires authentication and what type of authentication is required for any particular transaction is currently performed by the ActiveAccess system. However, in order to delegate any of these duties to external issuer systems, some level of integration will be required.

In order to determine the correct 3-D Secure registration status, an issuer may be required to maintain the status of each of its cardholder's within the ActiveAccess system. Where this requires a significant investment in the development of maintenance procedures, an alternative may be to connect to determine the registration status of a cardholder by connecting with an issuer's systems. This procedure will remove the need to synchronise systems and ensure the maintenance of only one source of truth.

In a similar way, where an issuer is currently providing authentication services for its cardholders and wishes to re-use some of these services, it is possible to connect to the issuer's existing authentication system and leverage off their existing process for cardholder authentication. While the capability to perform second factor authentication is provided by ActiveAccess, this integration may be the issuer's preferred implementation model. This approach ensures a seamless user experience for the bank's customers across its banking channels.

The following sections explain the messaging requirements for connecting between ActiveAccess and an issuer's remote systems.

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Message Format

SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services for messaging between ActiveAccess and an external system. It relies on Extensible Markup Language (XML) for its message format, and usually relies on other Application Layer protocols, most notably Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

The Web Services Description Language is an XML-based language that is used for describing the functionality offered by a Web service. A WSDL description of a web service (also referred to as a WSDL file) provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns. It thus serves a roughly similar purpose as a method signature in a programming language.

Support for generating client-side and server-side API code based on WSDL is provided in most languages.

For WSDL of the services discussed in this document, refer to **Remote System Integration WSDL**.

Request

During a 3-D Secure transaction, the first stage of the messaging is to determine the registration status of a cardholder. By integrating with the issuer's system, ActiveAccess can determine the cardholder's registration status and therefore determine whether a transaction requires authentication.

Having determined a transaction requires authentication, the second stage in the process is to perform an authentication and by integrating with the issuer's system, ActiveAccess is capable of reusing the issuer's infrastructure to determine the authentication result. At the end of the process, the ActiveAccess system responds to the MPI with the authentication result in accordance with the 3-D Secure protocol.

The purpose of remote system integration is to:

- Determine the registration status of a cardholder

  and/or

- Initiate and verify the cardholder authentication.

The types of messages sent by ActiveAccess are:

- **VerifyRegistration**: determine the registration status of a cardholder
- **InitAuthentication**: initiate the cardholder authentication process
- **VerifyAuthentication**: verify the authentication result
- **PreAuthentication**: determine the action for exemption
- **VerifyIdentity**: verify the identification results
- **Register**: register the card
- **ResetPassword**: initiate the reset password process
- **Ping**: determine the status of the service.

Note

Messages sent between ActiveAccess and the remote system for this purpose do not carry any session information and therefore are considered to be stateless

CAAS Services

Table 1 - CAAS Services

CAAS Service	Table 1	
Operation	**Description**	**Usage**
VerifyRegistration	Used to verify the registration status of a cardholder.	Required for a verify registration request
InitAuthentication	Used to initiate the authentication process for out-of-band authentication.	Required for an initiate authentication request
VerifyAuthentication	Used to determine the authentication result.	Required for a verify authentication request
PreAuthentication	Used to determine the action for exemption	Optional

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

CAAS Service	Table 1	
VerifyIdentity	Used to verify the identification results	Required for a reset password request and register request
Register	Used to register the card	Required for a register request
ResetPassword	Used to initiate the reset password process	Required for a reset password request
Ping	Used to determine if service is up and running	Optional

Verify Registration

The Verify Registration request is used to determine the registration status of a cardholder, within the remote system. Where a cardholder cannot be uniquely identified, such as the case where primary and secondary exist, it may be necessary for the remote system to provide the registration status of all related cardholders. ActiveAccess will then determine the appropriate course of action based on the response and in line with the issuer's business requirements.

Once the cardholder has been uniquely identified and where authentication is required, ActiveAccess should commence the appropriate authentication process.

Verify Registration Request

Table 2 - VerifyRegReq

VerifyRegReq	Table 2		
Attribute	**Description**	**Usage**	Sample Value
Card	Refer to *Table 3 - Card*	Required	
Transaction	Additional transaction information may include transaction; cardholder and merchant information such as **MerchantID** and **AcqBIN**. Refer to *Table 4 - Transaction*.	Optional	

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

VerifyRegReq	Table 2		
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optional- If present, it means that ActiveAccess has generated an IV parameter and critical card information has been encrypted using the CBC mode and the generated IV, otherwise ECB or plain mode has been used instead. 16 bytes when AES, 8 bytes when DESede.	8F51F71064DB2B65

Table 3 - Card

Card	Table 3		
Attribute	**Description**	**Usage**	**Sample Value**
ID	A unique cardholder identifier	Optional. Up to 2000 characters.	2345678901
Number	Card number (If an encryption KeyStore has been defined for the issuer or group of issuers, card number will be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode and IV, then HEX encoded and included in the message. Therefore, the CAAS server will need to decrypt this field using DESede/CBC/PKCS5Padding mode and the request's IV before using it in the process)	Optional. Up to 64 characters.	5012345678901234

Card	Table 3		
CardName	Name on card (If an encryption KeyStore has been defined for the issuer or group of issuers, name on card will be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode and IV, then HEX encoded and included in the message. Therefore, the CAAS server will need to decrypt this field using DESede/CBC/PKCS5Padding mode and the request's IV before using it in the process)	Optional. Up to 512 characters.	JOE CITIZEN
Type	Card type	Optional. Up to 3 characters. Valid types: VbV – Visa, SPA – Mastercard, JCB – JCB, SK – American Express, DC - Diners Club International.	SPA
Context_Blob	A context detail that may be used in subsequent calls	Optional. This field can be ignored by CAAS in VerifyRegReq as ActiveAccess does not use it and only echoes it in InitAuthReq and VerifyAuthReq if it has been set in VerifyRegResp.CardInfo. Context_Blob by CAAS. Length not defined.	12345678901235467890
LanCode	A code between 0 to 4 that presents the cardholder's preferred language	Optional. 1 character in length.	0

Table 4 - Transaction

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Transaction	Table 4		
Attribute	**Description**	**Usage**	**Sample Value**
XID	The transaction ID as defined in the PAReq message	Optional. Up to 28 characters.	MDAwMDAwMDAwMDAwMDAwMDAxMDA=
PurchaseDate	The transaction purchase date and time as defined in the PAReq message	Optional. Up to 17 characters in XMLGregorianCalendar format.	20091023 06:11:00
PurchaseAmount	The transaction purchase amount as defined in the PAReq message	Optional. Up to 12 characters in decimal format.	12345
PurchaseCurrency	The 3-digit transaction currency value as defined in the PAReq message. Refer to **Country and Currency Codes**	Optional. Up to 3 digits.	840
PurchaseExponent	The minor units of currency specified in ISO 4217	Optional. 1 character in length.	2
PurchaseDesc	A description of the purchase as defined in the PAReq message	Optional. Up to 125 characters.	Blue Shirt
MerchantID	The merchant ID as defined in the PAReq message	Optional. Up to 24 characters.	123456789012345
AcqBIN	The acquirer BIN as defined in the PAReq message	Optional. Up to 11 characters.	412345

Transaction	Table 4		
MerchantName	The merchant name as defined in the PAReq message	Optional. Up to 25 characters.	Test Merchant
MerchantURL	The fully qualified merchant URL as defined in the PAReq message	Optional. Up to 2048 characters.	http://www.testmerchant.com.au/
MerchantCountry	The 3-digit merchant country code as defined in the PAReq message. Refer to **Country and Currency Codes**	Optional. 3 digits in length.	036
CardExpiry	The 4-digit expiry date of the card as defined in the PAReq message, e.g. YYMM	Optional. 4 or 6 digits in length.	1012
CardholderIP	The IP address of the cardholder browser where available	Optional. 15 or 45 characters in IPv4 or IPv6 format.	192.168.0.157
CVD	Card Verification Data code is the 3 or 4-digit code found on the back of a payment card	Optional. 3 or 4 digits in length.	0320
issuerName	Name of Issuer/Bank to be displayed on OOB page	Optional. Up to 64 characters.	Any Bank
theeDSProtocolVersion	Version of 3DS protocol in x.x.x format	Optional. 5 characters in length.	2.1.0

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Transaction	Table 4		
+ acsTransId	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.	Optional. 36 alphanumeric characters in length.	ee5de3bc-a1a3-4648-9c5f-350422146fe1
+ threeDSTransId	Universally Unique transaction identifier assigned by the 3DS Server to identify a single transaction.	Optional. 36 alphanumeric characters in length.	we5de3bc-a213-46lk-9cas-35456ed46fe1
+ dsTransId	Universally Unique transaction identifier assigned by the Directory Server to identify a single transaction.	Optional. 36 alphanumeric characters in length.	tg6de3bc-a213-4r3k-9c12-35456ed4edr43

Verify Registration Response

A response message should be sent back for each request. The response message should provide the result of the request message with details of appropriate response information or errors as appropriate. Where one card is found in the remote system, registration details for that card should be included in the response. Where multiple cards are found, the registration details for each of the cards should be included in the response.

Table 5 - VerifyRegResp

VerifyRegResp	Table 5		
Attribute	**Description**	**Usage**	**Sample Value**

VerifyRegResp	Table 5		
CardInfo	If the request was successful and at least one card record was found, card related data may include primary/secondary cardholder indicator, registration status, authentication required indicator, authentication type, a card identifier and a SIS data. Refer to *Table 6 - CardInfo*.	Conditional. If response code is not presented, at least one CardInfo should exist.	
Code	Response code: 0 - request was successful but no card records were found 1 - request has been successfully processed but there are warnings (NOTE- Please see below) 2 - error in processing the request.	Required. Included where no card records are found or an error occurred.	0
ErrorMessage	A descriptive message that identifies the category of the error	Conditional. Included where a Code is returned in the response.	No card(s) found
ErrorDetail	A more detailed description of the error	Conditional. Included where a Code is returned in the response.	No card(s) matching the request were found

Note

ActiveAccess treats warnings (code=1) as errors unless the exact **ErrorMessage** is introduced in **AA_HOME/caaswarning.properties** with a code less than 2000. Changing this file requires a restart to take effect.

Table 6 - CardInfo

CardInfo	Table 6		
Attribute	**Description**	**Usage**	**Sample Value**
CardID	A unique cardholder identifier to be used as the value of the Card.ID attribute in subsequent request messages	Conditional. At least one of the Context_Blob or CardID is required. ActiveAccess echoes the Context_Blob into both Card.ID and Card. Context_Blob of the subsequent InitAuthReq and VerifyAuthReq if no CardID is returned by CAAS	2345678901

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

CardInfo	Table 6		
Card Name	Cardholder name to be used for specifying the exact cardholder when there are multiple cardholders for an identical card number (If an encryption KeyStore has been defined for the issuer or group of issuers, cardholder name must be encrypted using DESede/CBC/PKCS5Padding mode and message request IV by CAAS server, then HEX encoded and included in the message. ActiveAccess will decrypt this field using DESede/CBC/PKCS5Padding mode and the message request IV before using it in the process.)	Optional	John Smith

CardInfo	Table 6		
PAM	Personal Assurance Message (If an encryption KeyStore has been defined for the issuer or group of issuers, PAM must be encrypted using DESede/CBC/PKCS5Padding mode and the message request IV by the CAAS server, then HEX encoded and included in the message. ActiveAccess will decrypt this field using DESede/CBC/PKCS5Padding mode and the message request IV before using it in the process.)	Optional	This is my Bank
Context_Blob	A context detail that may be used in subsequent calls	Conditional. At least one of the Context_Blob or CardID is required. ActiveAccess echoes the Context_Blob into both Card.ID and Card. Context_Blob of the subsequent InitAuthReq and VerifyAuthReq if no CardID is returned by CAAS	12345678901234567890

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

CardInfo	Table 6		
Prisec	Primary or Secondary Cardholder 1 - Primary 2 - Secondary	Conditional	1
RegStatus	Registration Status: 1 - Enrolled (ActiveAccess enrolment status of pre-registered) 2 - Registered (ActiveAccess enrolment status of registered) 3 - Locked 4 - Unknown 5 - Error 6 - Temporarily Exempt 7 - Permanently Exempt 8 - Lost 9 - Stolen 10 -Restricted 11 - Card Number Error 12 - No Account 13 - Fraud 14 - Expired	Conditional. If SIS has data, RegStatus will not be considered.	2

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

CardInfo	Table 6		
AuthRequired	Authentication Required: 1 - Yes 2 - No	Conditional. If SIS has data, AuthRequired will not be considered.	1
AuthType	Authentication Type: 1 - Password 2 - SMS 3 - OTP device 4 - Virtual OTP device 5 - CAP/DPA 6 - Verify by Voice 7 - USS 8 - Q&A 9 - OLB 10 - CR 11 - BIO 12 - PKI 13 - TTP 14 - Email 15 - OOB	Conditional. If SIS has data, AuthType will not be considered	1

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

CardInfo	Table 6		
RegToken	The variable part of a message to be displayed to user/cardholder in the registration page. It reflects the number of times the cardholder opts-out during the registration process.	Optional. e.g. CAAS server wants to limit the number of times that a user/cardholder can opt-out from the registration process.	3 (e.g. of the message in the registration page: You have opted-out of the registration process 3 times)
AuthTypeSup	Supplementary authentication types that user/cardholder's account supports: 1 - Password 2 - SMS 3 - OTP device 4 - Virtual OTP device 5 - CAP/DPA 6 - Verify by Voice (OOB Biometrics) 7 - USS 8 - Q&A 9 - OLB (OOB Login) 10 - CR 11 - BIO (OOB Biometrics) 12 - PKI 13 - TTP (OOB Other) 14 - Email 15 - OOB (OOB Other)	Optional. More than one supplementary authentication type can be set for the authentication page to be selected by user/cardholder during the authentication	2, 3

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

CardInfo	Table 6		
SIS	Refer to *Table 7 - SIS*	Conditional. If exists, it takes precedence over RegStatus, AuthRequired and AuthType	
ProofAttempt	The availability of the Opt-Out option, as opposed to Cancel, for the cardholder. True - request identification parameters False - Proof of Attempt disabled,Opt -Out option not available **Note** - it is recommended to set this through ACS via **MIA > Issuers > Settings** instead of this parameter	Optional	false

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

CardInfo	Table 6		
ActivationDuringShopping	The ability to authenticate an enrolled cardholder by ID details for verification. True - request identification parameters False - registration pages are processed as without activation **Note** - it is recommended to set this through ACS via **MIA > Issuers > Settings** instead of this parameter	Optional	true
LanCode	The code of the preferred language saved for the cardholder. The value can be a digit between 0 to 4.	Optional	0 - default language 1 - 2nd language 2 - 3rd language 3 - 4th language 4 - 5th language

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

CardInfo	Table 6		
IdentityData	Attributes of Data: **Name** (required) - the name of the AuthData parameter to be used for data collection on the page **AuthType** (conditional) - empty value Format (optional) - the regular expression for verifying the value collected from the page **Mask** (optional) - True - input on the page will be masked False - input on the page will be in plaintext **Confirm** (optional) - True - an additional input field will be added to the page for confirmation False - no confirmation input field will be displayed on the page Refer to Table 8 - Data	Conditional. Required if ActivationDuringShopping is TRUE and RegStatus is 1. If RegStatus is not 1 and IdentityData has been returned, it will be used in the ResetPassword process.	identityData=[data={[value=<(null)>, error=<(null)>, name=pin, authType=<(null)>, format=\w+, mask=true, confirm=<(null)>]

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

CardInfo	Table 6		
twoFA	The availability of 2FA authentication option. 2FA authentication is a combination of: **Knowledge**: something only the user knows (e.g. password, pin, ID number) + **Ownership**: something only the user possesses (e.g. mobile device, token, smart card) or **Inherence**: something only the user is (e.g. fingerprint, face or voice recognition) The first factor must be knowledge; the second factor can be ownership or inherence.	Optional	true - enable two-factor authentication false - disable two factor authentication

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Table 7 - SIS

SIS	Table 7		
Attribute	**Description**	**Usage**	**Sample Value**
AccountState	Account State: 1 - Operational 2 - Unknown	Required	1
OperationalState	Operational State: 1 - Operational 2 - Locked Blank -Not Specified	Required	1
SecurityDeviceType	Security Device Type: 1 - Hard Token 2 - Soft Token 3 - SMS 4 - PIQ 5 - Email Blank - Not specified	Required	3
IsExempt	Authentication Exemption: True False Blank - Not specified	Required	2
IsPermanent	Permanent Authentication Exemption: True False Blank - Not specified	Required	2

Table 8 - Data

Data	Table 8		
Attribute	**Description**	**Usage**	**Sample Value**
Value	The value of Data	Optional	123456
Error	Refer to *Table 9* - Error	Optional	

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Table 9 - Error

Error	Table 9		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the request was successful 1 - there was an error and ActiveAccess should send the request again 2 - there was an error and ActiveAccess should cancel the authentication	Required	2
Message	A descriptive message that identifies the category of the error	Optional	No card(s) found
Detail	A more detailed description of the error	Optional	No card(s) matching the request were found

Pre Authentication

The ability to integrate ActiveAccess with an external risk engine has been established in the Pre Authentication process in which header data including cookie and HTTP header data in addition to potential extension information will be sent to CAAS for it to determine if authentication is required or exempt.

Pre Authentication

Table 10 - PreAuthReq

PreAuthReq	Table 10		
Attribute	**Description**	**Usage**	**Sample Value**
Card	Where a value for Card.ID or Context_Blob was returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request Refer to *Table 3 - Card*.	Required Either ID or Number and Type should be presented	card=[id=4564260131003313, number=4564-26XX-XXXX-3313, type=VbV, cardName=<(null)>, Context_Blob=595],
Transaction	Where messaging commences after the ActiveAccess system receives the PAReq, additional transaction, cardholder and merchant information is available. This information may be additionally sent to the issuer system for analysis and fraud detection purposes. Where required, the following data fields may be sent to the issuer's system in any of the request messages. Refer to *Table 4 - Transaction*.	Optional	transaction=[xid=MDAwMDAwMDAwMDAwMDAwMDAxMDA=, purchaseAmount=12365, purchaseCurrency=840, purchaseDate=[eon=<(null)>, year=2016, month=11, day=3, timezone=210, hour=10, minute=16, second=46, fractionalSecond=0.000],

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

PreAuthReq	Table 10		
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optional. If present, it means that ActiveAccess has generated an IV parameter and critical card information has been encrypted using the CBC mode and the generated IV, otherwise ECB or plain mode has been used instead.	8F51F71064DB2B65

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

PreAuthReq	Table 10		
HeaderParams	Attributes of Param: **Value** (required) **Key** (required) **Cookie** (optional)	Optional	headerParams=[param={[value=value1, key=key1, cookie=true],...}],
ExtensionParams	Attributes of Param: **Value** (required) **Key** (required)	Optional	extensionParams =[param={[value=value1, key=key1],...}],
AdditionalParams	Attributes of Param: **Value** (required) **Key** (required)	Optional	additionalParams =[param={[value=50, key=giftCardAmount],...}],

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Table 11 - HeaderParams

Key	Description	Sample Value
User-Agent	Either value of http request header parameter or browserUserAgent element of AReq	Mozilla/5.0 (X11; Linux x86_64; rv: 12.0) Gecko/20100101 Firefox/ 12.0
Accept	Either value of http request header parameter or browserAcceptHeader element of AReq	text/html
Accept-Language	Either value of http request header parameter or browserLanguage element of AReq	en-US
proxy-ip	Either value of http request header parameter or browserIp element of AReq	192.168.1.138
browserJavaEnabled	browserJavaEnabled element of AReq	true
browserTZ	browserTZ element of AReq	--
browserLanguage	browserLanguage element of AReq	en-US
deviceInfo	deviceInfo element of AReq	--

ExtensionParams

The elements differ by different message extensions are difend in messages. Like for intance, American Express extension params differ from MasterCard extension params.

Table 12 - AdditionalParams

Key	Description	Sample Value
shipAddrState	shipAddrState element of AReq	--
shipAddrCity	shipAddrCity element of AReq	--
shipAddrCountry	shipAddrCountry element of AReq	--
shipAddrLine1	shipAddrLine1 element of AReq	--

Key	Description	Sample Value
shipAddrLine2	shipAddrLine2 element of AReq	--
shipAddrLine3	shipAddrLine3 element of AReq	--
shipAddrPostCode	shipAddrPostCode element of AReq	--
billAddrState	billAddrState element of AReq	--
billAddrCity	billAddrCity element of AReq	--
billAddrCountry	billAddrCountry element of AReq	--
billAddrLine1	billAddrLine1 element of AReq	--
billAddrLine2	billAddrLine2 element of AReq	--
billAddrLine3	billAddrLine3 element of AReq	--
billAddrPostCode	billAddrPostCode element of AReq	--
deliveryEmailAddress	deliveryEmailAddress element of AReq	--
deliveryTimeframe	deliveryTimeframe element of AReq	--
giftCardAmount	giftCardAmount element of AReq	--
giftCardCount	giftCardCount element of AReq	--
giftCardCurr	giftCardCurr element of AReq	--
preOrderDate	preOrderDate element of AReq	--
preOrderPurchaseInd	preOrderPurchaseInd element of AReq	--
reorderItemsInd	reorderItemsInd element of AReq	--
shipIndicator	shipIndicator element of AReq	--
threeDSReqAuthData	threeDSReqAuthData element of AReq	--

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Key	Description	Sample Value
threeDSReqAuthMethod	threeDSReqAuthMethod element of AReq	--
threeDSReqAuthTimestamp	threeDSReqAuthTimestamp element of AReq	--
threeDSReqPriorAuthData	threeDSReqPriorAuthData element of AReq	--
threeDSReqPriorAuthMethod	threeDSReqPriorAuthMethod element of AReq	--
threeDSReqPriorAuthTimestamp	threeDSReqPriorAuthTimestamp element of AReq	--
threeDSReqPriorRef	threeDSReqPriorRef element of AReq	--
chAccAgeInd	chAccAgeInd element of AReq	--
chAccChange	chAccChange element of AReq	--
chAccChangeInd	chAccChangeInd element of AReq	--
chAccDate	chAccDate element of AReq	--
chAccPwChange	chAccPwChange element of AReq	--
chAccPwChangeInd	chAccPwChangeInd element of AReq	--
nbPurchaseAccount	nbPurchaseAccount element of AReq	--
provisionAttemptsDay	provisionAttemptsDay element of AReq	--
txnActivityDay	txnActivityDay element of AReq	--
txnActivityYear	txnActivityYear element of AReq	--
paymentAccAge	paymentAccAge element of AReq	--
paymentAccInd	paymentAccInd element of AReq	--
shipAddressUsage	shipAddressUsage element of AReq	--
shipAddressUsageInd	shipAddressUsageInd element of AReq	--

Key	Description	Sample Value
shipNameIndicator	shipNameIndicator element of AReq	--
suspiciousAccActivity	suspiciousAccActivity element of AReq	--

Pre Authentication Response

A response message should be sent back by the remote authentication system to decide on the continuation of the authentication process.

Table 13 - PreAuthResp

PreAuthResp	Table 13		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - The authentication will be exempted. The authentication will not be displayed and the appropriate response will be returned. 1 - The transaction is not exempt. The authentication page will be displayed. 2 - There was an error but ActiveAccess will display the authentication page and let the authentication continue. ActiveAccess will not cancel the authentication. 3 - The transaction is deemed to be high risk, ActiveAccess will decline the transaction.	Required	2

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

PreAuthResp	Table 13		
AuthType	The comma separated list of decided authTypes by risk engine integration: 1- Password 2- SMS 3- OTP device 4- Virtual OTP device 5- CAP/DPA 6- Verify by Voice 7- USS 8- Q&A 9- OLB 10- CR 11- BIO 12- PKI 13- TTP 14- Email 15- OOB	Optional	2, 14
ErrorMessage	A descriptive message that identifies the category of the error	Optional	No card(s) found
ErrorDetail	A more detailed description of the error	Optional	No card(s) matching the request were found

Initiate Authentication

The Initiate Authentication step is optional and depends upon the type of authentication device being used. Once the registration status of the cardholder has been determined, ActiveAccess may initiate the authentication process by sending a request to the issuer's remote system. This step may be used for the first, and subsequent, generate challenge requests.

This step will commonly be used to initiate out of band authentication such as SMS, Question and Answer, Challenge and Response and Email.

⚠ **Warning**

This messaging is generally used only for out of band authentication and may be initiated either automatically by the system or manually, such as when a cardholder clicks on a "Send SMS" button on the page.

Initiate Authentication Request

Table 14 - InitAuthReq

InitAuthReq	Table 14		
Attribute	**Description**	**Usage**	**Sample Value**
Card	Where a value for Card.ID or Context_Blob was returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card*.	Required. Either ID or Number and Type should be presented	card=[id=4564260131003313, number=4564-26XX-XXXX-3313, type=VbV, cardName=< null >, Context_Blob=595],
Transaction	Where messaging commences after the ActiveAccess system receives the PAReq, additional transaction, cardholder and merchant information is available. This information may be additionally sent to the issuer system for analysis and fraud detection purposes. Where required, the following data fields may be sent to the issuer's system in any of the request messages. Refer to *Table 4 - Transaction*.	Optional	transaction=[xid=MDAwMDAwMDAwM, purchaseAmount=12365, purchaseCurrency=840, purchaseDate=[orig_eon=< null >, orig_year=2016, orig_month=11, orig_day=3, orig_hour=10, orig_minute=23, orig_second=19, orig_fracSeconds=0.000, orig_timezone=210, eon=< null >, year=2016, month=11, day=3, timezone=210, hour=10, minute=23, second=19, fractionalSecond=0.000],

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

InitAuthReq	Table 14		
SMS	**Template** The SMS message to be sent to the cardholder populated with Transaction.MerchantName, Transaction.PurchaseAmount and Transaction.PurchaseCurrency in the format that is required by SMS Gateway to send to customer mobile. template = "Sample message here. Your OTP is {0}". **Notes:** 1. The {0} is the placeholder where CAAS injects the actual 6-digit OTP. 2. {0} can be anywhere in the template – the above is just an example. 3. The length of the text can be up to 160 chars (note, the {0} placeholder will expand from 4 characters to 6 characters, so free text is effectively 154 characters.)	Conditional, where the authentication channel is SMS. Up to 154 characters.	Your OTP is :{0} \r\n merName: Test Merchant, purchaseAmount: 123.65

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Email	Contains **Content**, **Subject** and the **Content-Type** of the email. **Content** (Required)- The content of the email to be sent to the cardholder, which can be populated with Transaction.MerchantName, Transaction.PurchaseAmount, Transaction.PurchaseCurrency, and any other information in the format that is configured by the bank to send to the customer's email address. **Notes:** 1. The {0} is the placeholder where CAAS injects the actual 6-digit OTP. 2. {0} can be anywhere in the template – the above is just an example. 3. The length of the text can be up to 160 chars (note, the {0} placeholder will expand from 4 characters to 6 characters, so free text is effectively 154 characters.) **Subject** (Required) - The subject of the email to be sent to the cardholder, which can be populated with *Issuer Name,* to send to the customer's email address. **Content-Type** (Required) - The content type of the email to be sent to the cardholder. This can be TEXT/PLAIN or TEXT/HTML.	Conditional. Content up to 1024 characters. Subject up to 998 characters. Content-Type up to 25 characters.

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

InitAuthReq	Table 14		
OobInfo	**Template** The message to be sent to the OOB application populated with Transaction. MerchantName, Transaction.PurchaseAmount and Transaction.PurchaseCurrency in the format that is required by OOB adapter to send to OOB. template = ": "$ThreeDSServerTransID", "purchaseAmount": "$PurchaseAmount", "purchaseCurrency": "$PurchaseCurrency", "purchaseExponent": "$PurchaseExponent", "messageCategory": "$MessageCategory", "deviceChannel": "$DeviceChannel", "acctNumber": "$AcctNumber", "merchantName": "$MerchantName", "cardHolderInfo": { "cardholderName": "$CardholderName", "email": "$Email", "homePhone": { "cc": "$HomePhone_cc", "subscriber": "$HomePhone_subscriber" }, "mobilePhone": { "cc": "$MobilePhone_cc", "subscriber": "$MobilePhone_subscriber" }, "shipAddrCity": "$ShipAddrCity", "shipAddrCountry": "$ShipAddrCountry", "shipAddrLine1": "$ShipAddrLine1", "shipAddrLine2": "$ShipAddrLine2", "shipAddrLine3": "$ShipAddrLine3", "shipAddrPostCode": "$ShipAddrPostCode", "shipAddrState": "$ShipAddrState", "workPhone": { "cc": "$WorkPhone_cc", "subscriber": "$WorkPhone_subscriber" } } }	Conditional. Where the authentication channel is any of OOB 6 - Verify by Voice 7 - USS 9 - OLB 11 - BIO 13 - TTP 15 - OOB. Up to 4000 characters.	{ "threeDSServerTransID": "$ThreeDSServerTransID", "purchaseAmount": "123", "purchaseCurrency": "840", "purchaseExponent": "2", "messageCategory": "01", "deviceChannel": "01", "acctNumber": "4123XXXXXXXX45", "merchantName": "Tet Merchant", "cardHolderInfo": { "cardholderName": "John", "email": "email@example.com", "homePhone": { "cc": "1", "subscriber": "530123112345" }, "mobilePhone": { "cc": "55", "subscriber": "23451443212" }, "shipAddrCity": "$ShipAddrCity", "shipAddrCountry": "$ShipAddrCountry", "shipAddrLine1": "$ShipAddrLine1", "shipAddrLine2": "$ShipAddrLine2", "shipAddrLine3": "$ShipAddrLine3", "shipAddrPostCode": "$ShipAddrPostCode", "shipAddrState": "$ShipAddrState", "workPhone": { "cc": "$WorkPhone_cc", "subscriber": "$WorkPhone_subscriber" } } }

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

InitAuthReq	Table 14		
AuthType	Authentication Type that cardholder requests to (re)initiate the one-time passcode for authentication: 2 - SMS 6 - Verify by Voice 7 - USS> 10 - CR 11 - BIO 14 - Email 15 - OOB	Conditional. Up to 2 characters.	2

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

InitAuthReq	Table 14		
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optional. If present, it means that ActiveAccess has generated an IV parameter and critical card information has been encrypted using the CBC mode and the generated IV, otherwise ECB or plain mode has been used instead. 8 or 16 characters in length.	8F51F71064DB2B65

Initiate Authentication Response

A response message should be sent back by the remote authentication system to indicate the status of sending an SMS or Email, or otherwise return AuthData for the authentication initiation.

Table 15 - InitAuthResp

InitAuthResp	Table 15		
Attribute	**Description**	**Usage**	**Sample Value**

InitAuthResp	Table 15		
Code	Response code: 0 - the request was successful 1 - there was an error and ActiveAccess should send the request again 2 - there was an error and ActiveAccess should cancel the authentication.	Required	2
ErrorMessage	A descriptive message that identifies the category of the error	Required	No card(s) found
ErrorDetail	A more detailed description of the error	Required	No card(s) matching the request were found

Verify Authentication

Where the remote system determines that authentication is required and after an authentication has been initiated, the cardholder should be presented with an appropriate page. In many circumstances this page will request the cardholder to enter their authentication credential, such as password or one-time password. However, in some circumstances, the screen presented may ask the cardholder to press a button after having completed their out of band authentication.

When a cardholder enters their password, ActiveAccess will format the details of the authentication request and send it to the remote system for verification. The response provided will determine the authentication status of the transaction, with ActiveAccess formatting the 3-D Secure payer authentication response message to be returned to the merchant's MPI.

Verify Authentication Request

Table 16 - VerifyAuthReq

VerifyAuthReq	Table 16	
Attribute	**Description**	**U**
Card	Where a value for Card.ID or Card.Context_Blob has returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card*.	F T
Token	The authentication number or password entered by the cardholder If an encryption KeyStore has been defined for the issuer or group of issuers, the token will be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode and IV, then HEX encoded and included in the message. CAAS server will need to decrypt this field using DESede/CBC/PKCS5Padding mode and the request's IV before using it in the process.	C M a c V t b C t

VerifyAuthReq	Table 16	
AuthData	Attributes of Data: Name (required) - the name of the AuthData parameter to be used for data collection on the page, AuthType (conditional) - the AuthType of AuthData which has an error, Format (optional) - the regular expression for verifying the value collected from the page, **Mask (optional)** - true - input on the page will be masked false - input on the page will be in plaintext, **Confirm (optional)** - true - an additional input field will be added to the page for confirmation f the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to *Table 8 - Data*.	(/

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

VerifyAuthReq	Table 16
Transaction	Additional transaction information may include transaction, cardholder and merchant information such as **XID, PurchaseDate**, **PurchaseAmount**, **PurchaseCurrency**, **PurchaseDesc**, **MerchantID**, **AcqBIN**, **MerchantName**, **MerchantURL**, **MerchantCountry**, **CardExpiry** and **CardholderIP** as described in the Initiate Authentication request section. Refer to *Table 4 - Transaction*.
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.
HeaderParams	Attributes of Param: **Value** (required) **Key** (required) **Cookie** (optional)
ExtensionParams	Attributes of Param: **Value** (required) **Key** (required)

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Verify Authentication Response

Table 17 - VerifyAuth

A response message should be sent back by the remote authentication system to indicate the success, or otherwise of the authentication verification.

VerifyAuthResp	Table 17		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the authentication was successful 1 - the authentication token was incorrect 2 - an error occurred and another attempt should be made 3 - the status of the card is locked 4 - an error occurred and no further attempts should be made. 5 - the interaction counter exceeded maximum interaction.	Required	3

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

AuthData	Attributes of Data: **Name** (required) - the name of the AuthData parameter to be used for data collection on the page **AuthType** (conditional) - the AuthType of AuthData which will be displayed on the authentication page **Format** (optional) the regular expression for verifying the value collected from the page **Mask** (optional) - true - input on the page will be masked false - input on the page will be in plaintext **Confirm** (optional) - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to *Table 8 - Data*.	Optional. When an error occurs, appropriate content can be returned. Otherwise, null will be returned.	authData=[data={[value=32132132, error=[code=1, message=value mismatch, detail=value mismatch], name=userId, authType=9, format= <(null)>, mask=<(null)>, confirm=<(null)>],[value=321321, error=[code=1, message=value mismatch, detail=value mismatch], name=password, authType=9, format=<(null)>, mask=<(null)>, confirm=<(null)>]}]

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

VerifyAuthResp	Table 17		
ErrorMessage	A descriptive message that identifies the category of the error	Required	Card is locked
ErrorDetail	A more detailed description of the error	Required	The status of card is locked due to multiple unsuccessful login tries.
HeaderParams	Attributes of Param: Value (required) Key (required) Cookie (optional)	Optional	headerParams=[param={[value=value1, key=key1, cookie=true],...}],
ExtensionParams	Attributes of Param: Value (required) Key (required).	Optional	extensionParams=[param={[value=value1, key=key1],...}],

Verify Identity

Verify Identity data is used in ADS or the Forgot password process to primarily verify the identity of the cardholder before changing/setting authentication data.

A request message should be sent to CAAS with the user identity data and to have CAAS verify the data.

Verify Identity Request

Table 18 – VerifyIdentityReq

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

VerifyIdentityReq	Table 18		
Attribute	**Description**	**Usage**	**Sample Value**
Purpose	An attribute which indicates if Identity data are for the ADS or Forgot password process. 1 = reset password 2 = ADS	Required	1
IdentityData	Attributes of Data: **Name** (required) - the name of the IdentityData parameter to be used for data collection on the page **AuthType** (conditional) - empty value **Format** (optional) - empty value **Mask** (optional) - empty value **Confirm** (optional) - empty value Refer to **Table 8 - Data**.	Required	identityData=[data={[value=User1, error=<(null)>, name=cname, authType= <(null)>, format=<(null)>,, mask=<<(null)>,, confirm=<(null)>,], [value=123456, error=<(null)>,, name=pin, authType=<(null)>,, format=<(null)>,, mask=<(null)>, confirm=<(null)>,]}]
Transaction	Additional transaction information may include transaction, cardholder and merchant information such as **XID**, **PurchaseDate**, **PurchaseAmount**, **PurchaseCurrency**, **PurchaseDesc**, **MerchantID**, **AcqBIN**, **MerchantName**, **MerchantURL**, **MerchantCountry**, **CardExpiry** and **CardholderIP** Refer to *Table 4 - Transaction* for details.	Optional	

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

VerifyIdentityReq	Table 18		
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optional. If present, it means that ActiveAccess has generated an IV parameter and critical card information has been encrypted using the CBC mode and the generated IV, otherwise ECB or plain mode has been used instead.	8F51F71064DB2B65
Card	Where a value for Card.ID or Card. Context_Blob has been returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card* for details.	Required. Either ID or Number and Type should be presented.	card=[id=4564260131003313, number=4564-26XX-XXXX-3313, type=VbV, cardName=<(null)>, Context_Blob=595]

Verify Identity Response

A response message should be sent back to ActiveAccess to inform whether user identity has been verified or not and to return the reason in case of identity failure. In addition, it returns failed identity items to be highlighted in the page. In the case of a successful response, it returns AuthData to be asked for a subsequent authentication process.

Table 19 - VerifyIdentityResp

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

VerifyIdentityResp	Table 19		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the authentication was successful, 1 - the authentication token was incorrect, 2 - an error occurred and another attempt should be made, 3 - the status of the card is locked, 4 - an error occurred and no further attempts should be made.	Required	3

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

VerifyIdentityResp	Table 19		
IdentityData	Attributes of Data: Name (required) - The name of the IdentityData parameter to be used for data collection on the page, AuthType (conditional) - empty value, Format (optional) - the regular expression for verifying the value collected from the page, **Mask (optional)** - true - input on the page will be masked false - input on the page will be in plaintext **Confirm (optional)** - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to Table 8 - Data.	Optional. When an error occurs, appropriate content can be returned. Otherwise, null will be returned.	identityData=[data={[value=administrator11, error=[code=1, message=value mismatch, detail=value mismatch], name=cname, authType=<(null)>, format=<(null)>,, mask=<(null)>,, confirm=<(null)>],[value=123456, error=<(null)>, name=pin, authType=<(null)>, format=<(null)>, mask=<(null)>, confirm=<(null)>]}]

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

VerifyIdentityResp	Table 19		
AuthData	Attributes of Data: Name (required) - the name of the AuthData parameter which will be registered or reset for the card, AuthType (conditional) - the AuthType of AuthData which will be registered or reset for the card, Format (optional) - the regular expression for verifying the value collected from the page, **Mask (optional)** - true - input on the page will be masked false - input on the page will be in plaintext, **Confirm (optional)** - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to *Table 8 - Data*.	Required. If VerifyIdentityReq.purpose=1, reset AuthData will be returned. If VerifyIdentityReq.purpose=2, a list of all AuthData will be returned for the cardholder to choose from and register with.	authData=[data={[value=<(null)>, error=<(null)>, name=password, authType=1, format=<(null)>, mask=true, confirm=true],[value=<(null)>, error=<(null)>, name=mobileNo, authType=2, format=<(null)>, mask=<(null)>, confirm=true], [value=<(null)>, error=<(null)>, name=token, authType=2, format=<(null)>, mask=<(null)>, confirm=<(null)>] … }]
ErrorMessage	A descriptive message that identifies the category of the error	Optional	Card is locked

VerifyIdentityResp	Table 19		
ErrorDetail	A more detailed description of the error	Optional	The status of card is locked due to multiple unsuccessful login attempts.

Register

A request message should be sent to CAAS to set Authentication data for subsequent authentication.

Register Request

Table 20 – RegisterReq

RegisterReq	Table 20	
Attribute	**Description**	**Usag**
RegisterData	Attributes of Data: Name (required) - the name of the collected RegisterData from the page, AuthType (conditional) - the AuthType of the collected RegisterData from the page, Format (optional) - empty value, Mask (optional) - empty value, Confirm (optional) - empty value. Refer to *Table 8 - Data*.	Requ
Card	Where a value for Card.ID or Card. Context_Blob has returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card*.	Requ
Transaction	Additional transaction information may include transaction, cardholder and merchant information such as **XID, PurchaseDate**, **PurchaseAmount**, **PurchaseCurrency**, **PurchaseDesc**, **MerchantID**, **AcqBIN**, **MerchantName**, **MerchantURL**, **MerchantCountry**, **CardExpiry** and **CardholderIP.** Refer to *Table 4 - Transaction*.	Optio

RegisterReq	Table 20	
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optio prese mear Activ has gener IV pa and c card inforr has b encry using CBC r and tl gener other ECB (mode been inste

Register Response

Table 21 – RegisterResp

RegisterResp	Table 21		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the authentication was successful, 1 - the authentication token was incorrect, 2 - an error occurred and another attempt should be made, 3 - the status of the card is locked, 4 - an error occurred and no further attempts should be made.	Required	3
RegisterData	Attributes of Data: Name (required) - the name of the RegisterData parameter, **Format (optional)** - the regular expression of RegisterData for verifying the value which is collected from the page, **Mask (optional)** - true - input on the page will be masked false - input on the page will be in plaintext, **Confirm (optional)** - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page.	Optional. If an error occurs, registerData would be sent back to ACS with an appropriate error message. Refer to *Table 8 - Data*.	registerData=[data={ [value=<(null)>, error=[code=1, message=invalid, detail=invalid], name=password, authType=1, format=<(null)>, mask=<(null)>, confirm=<(null)>]}]

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

RegisterResp	Table 21		
ErrorMessage	A descriptive message that identifies the category of the error	Optional	
ErrorDetail	A more detailed description of the error	Optional	

Reset Password

Reset Password Request

A request message should be sent to CAAS for ResetPasswordData and have data set for further use.

Table 22 - ResetPasswordReq

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

ResetPasswordReq	Table 22
Attribute	**Description**
ResetPasswordData	Attributes of Data: Name (required) - the name of the ResetPasswordData parameter, collected from the page, AuthType (conditional) - the AuthType of ResetPasswordData, collected from the page, Format (optional) - empty value, Mask (optional) - empty value, Confirm (optional) - empty value Refer to *Table 21 – ResetPasswordResp*.
Card	Where a value for Card.ID or Card. Context_Blob has been returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card*.
Transaction	Additional transaction information may include transaction, cardholder and merchant information such as **XID, PurchaseDate**, **PurchaseAmount**, **PurchaseCurrency**, **PurchaseDesc**, **MerchantID**, **AcqBIN**, **MerchantName**, **MerchantURL**, **MerchantCountry**, **CardExpiry** and **CardholderIP**. Refer to *Table 4 - Transaction*.

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

ResetPasswordReq	Table 22
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.

Reset Password Response

A response message should be sent back to ActiveAccess to indicate the result of the reset password process in CAAS.

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Table 23 – ResetPasswordResp

ResetPasswordResp	Table 23		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the authentication was successful, 1 - the authentication token was incorrect, 2 - an error occurred and another attempt should be made, 3 - the status of the card is locked, 4 - an error occurred and no further attempts should be made.	Required	3

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

ResetPasswordResp	Table 23		
ResetPasswordData	Attributes of Data: Name (required) - the name of the ResetPasswordData parameter, **AuthType (conditional)** - the AuthType of ResetPasswordData, **Format (optional)** - the regular expression for verifying the value collected from the page, **Mask (optional)** - true - input on the page will be masked true - input on the page will be masked false - input on the page will be in plaintext, **Confirm (optional)** - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to **Table 8 - Data**.	Optional. When an error occurs, appropriate content can be returned.	resetPasswordData= [data={[value=<(null)>, error=[code=1, message=invalid, detail=invalid], name=password, authType=1, format=<(null)>, mask=<(null)>, confirm=<(null)>]}]
ErrorMessage	A descriptive message that identifies the category of the error	Optional	
ErrorDetail	A more detailed description of the error	Optional	

Ping

The ping request is used to determine the responsiveness and availability of the server. Simply send a ping request to the server to check if the service is up and operational or not.

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Ping Request

Ping has no request parameter.

Ping Response

Ping has no response. Successful return of the operation invocation without any exception means the service is up and running.

Messaging Requirements

Securing Message Channel

Communication security must be ensured by using SSL with server and client authentication.

Critical Card Data Encryption and Decryption

The key, which is used for encrypting/decrypting the critical card data, must be a 112 or 168 bit DESede key. A KeyStore with the following details should be prepared for the encryption key that is to be uploaded, through MIA, for the specified issuer or group of issuers:

KeyStore type/format: JCEKS

KeyStore provider: SunJCE

Key algorithm: DESede

Key size: 112 or 168 bit

Key name: can be any

No of keys in the KeyStore: Only one key must be populated in the KeyStore

Such KeyStores can be easily created through the Java keytool utility using the following command:

```
keytool -genseckey -alias enckey168 -keypass 123456 -keyalg DESede -keysize 168 -
keystore enc-key.JKS -storepass 123456 -storetype JCEKS
```

If IV is set for the request, the CAAS server needs to get the IV by HEX decoding and decrypting the VerifyRegReq.IV / InitAuthReq.IV / VerifyAuthReq.IV using the encryption key in DESede/ECB/

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

PKCS5Padding mode, before decrypting the critical card data in DESede/CBC/PKCS5Padding mode using the obtained IV from the request.

Calling Convention

Requests will be sent using SOAP on HTTPS.

Remote System Integration WSDL

Important

It is important to ensure messages conform to the requirements of the remote system integration API by validating them against the WSDL and XSD schema.

The Remote System Integration WSDL and XSD schema can be found in the ActiveAccess installation package in the following path:

```
ActiveAccess/files/acs.war/WEB-INF/lib/caas.client-*.jar
```