

CardEase 3DS Server

NMI Technical Documentation

Document Version: 1.0.5

EMV 3DS Version: 2.1.0

Date: 2022-12-19



Table of Contents

Table of Contents	1
Document Control	5
CardEase 3DS Server Integration	6
Introduction	6
Pre-requisites	7
Test Platform Merchant Details	7
Authentication Flow	9
Step-By-Step	9
Frictionless Flow	11
Challenge Flow	12
Authentication	13
Authentication Error	13
End-Points & Models	14
Test Domain	14
Live Domain	14
Cardholder Enrollment Check	15
Path	15
Models	15
Request	15
Response	15
Initialise Authentication	16
Path	16
Models	16

Request	16
Response	16
Authentication	18
End-Point	18
Models	18
Request	18
Top Level	18
Account Information	23
Authentication Information	26
Card Holder Information	27
Phone Number	31
Merchant Risk Indicator	32
Response	34
Retrieve Results (GET request)	36
Path	36
Example	36
Models	36
Request	36
Response	36
Event Callback	39
POST Parameters	39
Events	39
3DSMethodFinished	39
3DSMethodSkipped	39
InitAuthTimedOut	40

AuthResultReady	40
Callback Example (PHP)	41
API Error Response	42
Error Response	42
Appendix A: API Values	44
TransStatus	44
TransStatusReason	45
Account Type	47
Authentication Indicator	48
AuthenticationType	49
ChallengeInd	50
TransType	51
ThreeDSReqAuthMethod	52
ShipIndicator	53
Error Codes	54
3DS Error Codes	54
Transaction Error Codes	56
General Error Codes	57
Appendix B: Glossary	58
Appendix C: Test Platform Details	61
Test Merchant	61
Test Cards	61
Expiry Date / Cardholder Name	61
Successful Frictionless Flow Authentication	62
Successful Challenge Flow Authentication	63

Authentication Attempted	64
Authentication Failed	65
Authentication Unavailable	66
Authentication Rejected	67

Document Control

Version	Author	Date	Pages Changed	Summary of Changes
1.0.0	Jarrett Chamberlin	2021-07-15	All	Initial Draft for EMV 3DS 2.1.0
1.0.1	Jarrett Chamberlin	2021-08-24	19	Added a required field to the Authentication call - transType is mandatory for Visa 3DS2 so this has been made clear.
1.0.2	Jarrett Chamberlin	2021-11-15	18, 27, 61	Made it clear that cardholderName is a required field within the Authentication models. Added a few notes regarding the cardcapitalisation differences between cardHolderInfo and the cardholderName contained within it. Test card section updated to make name and expiry date requirements more clear.
1.0.3	Jarrett Chamberlin	2021-03-21	15, 17, 20	Added details about how to perform 3DS2 authentications when using CardEase tokens (<i>cardGuid</i> / <i>cardHash</i>).
1.0.4	Jarrett Chamberlin	2022-10-26	44, 45	Update Events section to include information about 3DSMethodHasError event. Updated wording on InitAuthTimedOut event.

				Updated callback example with the two above events.
1.0.5	Jarrett Chamberlin	2022-12-19	31	Added a new note to the cardholder name field regarding the transliteration of non-ASCII characters.

CardEase 3DS Server Integration

Introduction

The CardEase 3DS Server is a hosted 3-D Secure Server (3DS Server) that allows for E-Commerce sites to accept and process Visa Secure, Mastercard Identity Check, Amex SafeKey 2.0 and Discover ProtectBuy 2.0 authentications.

The result of these authentications can then be used during the payment authorisation process providing conformance with card scheme rules and greater liability shift for the merchant.

There's a lot more additional data able to be collected as a part of the EMV 3DS flow. Many of these fields are gathered during the [Authenticate](#) step.

Most of these fields are not considered mandatory, however you are encouraged to include as much data as possible.

By providing more data, the issuer has more to draw upon when deciding whether they need to initiate a challenge step up or to allow the authentication process to follow the frictionless flow.

The CardEase 3DS Server makes use of the HTTP protocol using various methods (GET / POST) in order to perform cardholder authentication via the EMV 3DS 2.1 protocol.

Detailed below are the specifics on how to integrate with the CardEase 3DS Server in order to perform Browser Based EMV 3DS 2.1 authentication.

This includes going over the ideas of a “3DS Web Adapter” and a “3DS Requestor” which are considered two key parts of your integration.

Pre-requisites

This documentation assumes that you already have a merchant boarded onto our 3DS Server.

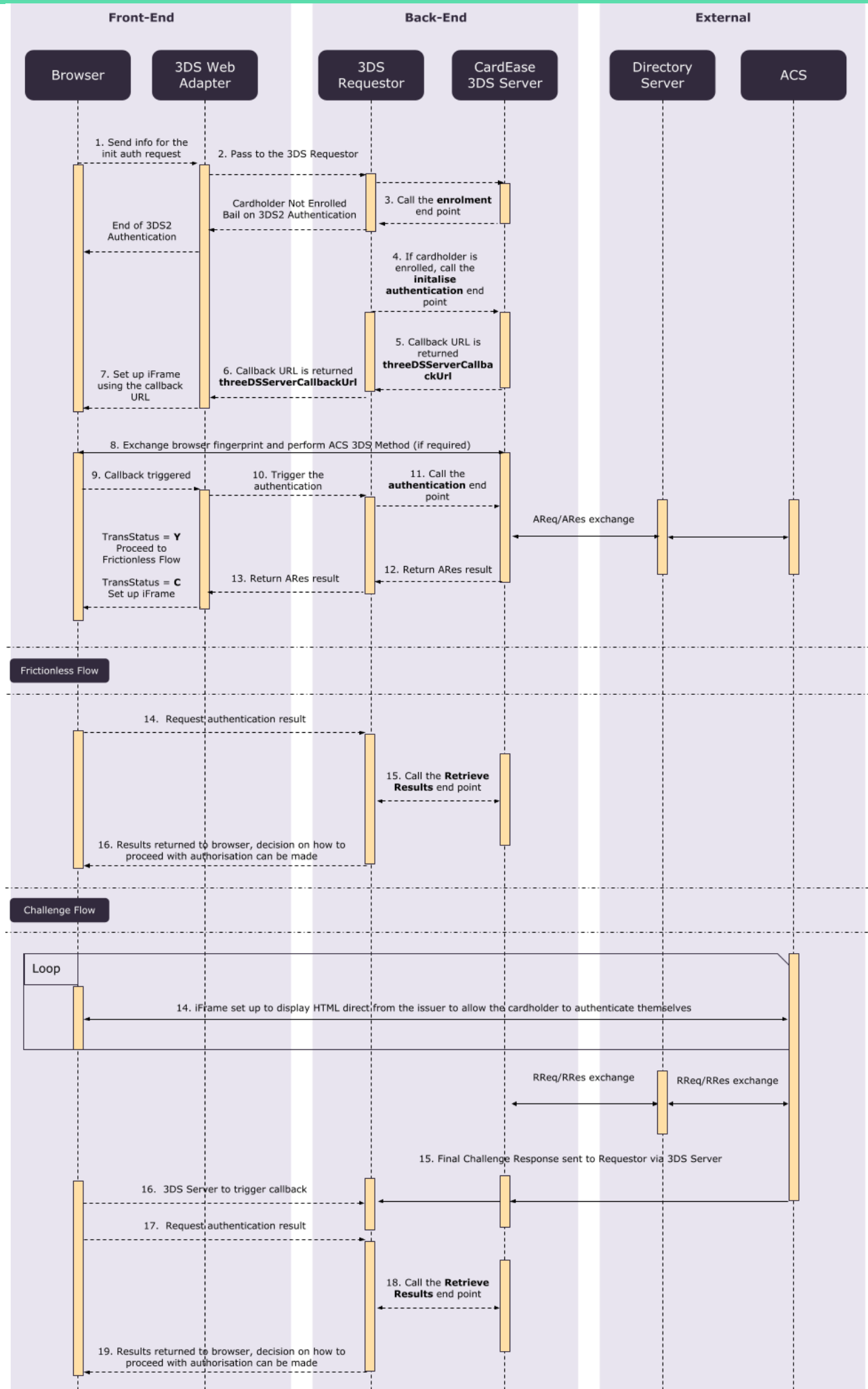
Once an EMV 3-D Secure enabled merchant has been boarded to CardEase, you'll need the following fields:

- Merchant ID
- Merchant Name
- Merchant Token

These can be supplied by NMI once the boarding has been completed.

Test Platform Merchant Details

For details of our test platform merchant details, please see [Appendix C](#).



Authentication Flow

Step-By-Step

1. The browser sends an “initialise authentication” request via the 3DS Web Adapter
 - The 3DS Web Adapter is a Javascript component/application written by the integrator that interacts with the 3DS Requestor via asynchronous calls from the browser.
2. The 3DS Web Adapter passes on the request to the 3DS Requestor.
 - The 3DS Requestor is a server-side component written by the integrator that receives requests from the 3DS Web Adapter and interacts with the 3DS Server.
3. The 3DS Requestor performs an enrolment check by sending a request to the [Cardholder Enrolment Check](#) end-point
 - If the card is not enrolled, the 3DS Requestor shall not continue with the 3DS2 authentication
4. The 3DS Requestor formulates and sends an “initialise authentication” request to the 3DS Server to the [Initialise Authentication](#) end-point
 - **Important:** eventCallbackUrl will be used by the 3DS Server in order to send requests from 3DS Server to the 3DS Requestor.
 - **Important:** authUrl is the URL to be used to execute the auth call
 - **Important:** monUrl is the URL to be used in the event of a timeout with the ACS. This should be set up in it's own iFrame.
5. The 3DS Server will respond with a `threeDSServerCallbackUrl` to the 3DS Requestor
6. The 3DS Requestor returns this Callback URL to the 3DS Web Adapter
7. The 3DS Web Adapter sets up an iFrame in the Browser with the returned callback URL as its source
8. The iFrame will automatically perform browser finger-printing and post the data to the necessary places
 - This will always be posted to the 3DS Server

- If there is a “3DS Method URL” assigned to the card range, this is the point at which any browser data will be posted to the necessary 3DS Access Control Server (3DS ACS)
9. Once fingerprinting has been completed, the 3DS Server will trigger a callback to the 3DS Requestor via the 3DS Web Adapter
- See [Event Callback](#) section below for what will be sent alongside the request here
 - **Important:** The `param` parameter value provided in the Event Callback contains some data that **must** be provided verbatim to the Authentication call in the next step.
10. The 3DS Web Adapter then sends the Authentication call to the 3DS Requestor
- This can be done via Javascript / AJAX
 - **Important:** The `browserInfo` field of this request **must** be provided as it is received in the previous step.
11. The 3DS Requestor then formulates the Authentication request to be sent to the 3DS Server via the [Authentication](#) end-point
- 3DS Server will perform the AReq / ARes part of the EMV 3DS process with the scheme directory servers
12. The 3DS Server will then send the results of the ARes call back to the 3DS Requestor, including a `transStatus` indicator.
13. The 3DS Requestor then sends this response back to the 3DS Web Adapter

What happens now depends on the value of `transStatus`.

If the value is **Y** - please view the steps for the Frictionless Flow below.

If the value is **C** - please view the steps for the Challenge flow below.

Frictionless Flow

14. The 3DS Web Adapter then sends a request to the 3DS Requestor with any required data to allow the 3DS Requestor to make a call to the Result end-point
15. The 3DS Requestor then formulates and sends a request to the [Retrieve Result](#) end point to retrieve the results of the authentication and return the results to the 3DS Web Adapter
16. The 3DS Web Adapter can then use the values returned as required. Please see the below [Retrieve Result response](#) model for more information on what can be returned here

NB: The 3DS Requestor may get all the result information as a part of the [Authentication](#) response. However, it's good practice to go back to the 3DS Server to get the results from the [Retrieve Result](#) end-point.

Challenge Flow

When you get a transStatus result of **C** from the Authenticate call, there will also be a **challengeUrl** field containing a URL.

14. The 3DS Web Adapter to set up another iFrame using the **challengeUrl** value as the **src**
 - This iFrame will allow the browser to connect directly to the Issuers ACS to allow the user to authenticate themselves
15. Once the user has authenticated themselves, the 3DS Server will receive the challenge result
 - This comes via the RReq/RRes messages sent by the ACS once the user has authenticated themselves
16. Once the user has authenticated themselves, the 3DS Server will call the Event Callback URL to trigger the next step with the 3DS Requestor
 - This will have the **AuthResultReady** event value when the user has finished
17. The 3DS Web Adapter then sends a request to the 3DS Requestor with any required data to allow the 3DS Requestor to make a call to the Result end-point
18. The 3DS Requestor then formulates and sends a request and then call the [Retrieve Result](#) end point to retrieve the results of the authentication and return the results to the 3DS Web Adapter
19. 3DS Web Adapter can then use the values returned as required. Please see the below [Retrieve Result response](#) model for more information on what can be returned here

Authentication

Header Name: **C3DS-Merchant-Token**

Header Value: Merchant token as provided after boarding, GUID format

To successfully authenticate with our hosted 3DS Server, you need to include the above header as a part of each request, with the merchant token that can be retrieved once the merchant has been boarded to the 3DS Server.

This token will only work when attempting authentications with the specific merchant for which it has been generated.

Authentication Error

In the event that you try to use a merchant token that doesn't match up with the boarded merchant, you'll receive the below HTML back, with the HTTP Code 401 (Unauthorized).

An example of such a response would be:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
  <title>401 Unauthorized</title>
</head>
<body>
  <h1>Unauthorized</h1>
  <p></p>
</body>
</html>
```

End-Points & Models

- All requests to use POST unless otherwise noted
- All requests/responses will be in JSON format and require the below header to be a part of each request
 - **Content-Type: application/json**
- All of the request/responses assume a HTTP Code of 200 to mean a successful request
- For error responses (HTTP Code 400), see the API Error Response section below
- Please ensure all requests have the authentication header as detailed in the Authentication section
- Fields formatted in **bold*** are considered mandatory fields and must be supplied with every request

Test Domain

<https://test3ds2s.cardeasxml.com:9603>

Live Domain

<https://3ds2s.cardeasxml.com:9603>

Cardholder Enrollment Check

Path

/api/v2/auth/enrol

Models

Request

Field	Value	Format
acctNumber*	Account number that will be used in the authorisation request for payment transactions.	Minimum Length: 13 Maximum Length: 19 Account number, numeric Mandatory when not using CardEase tokens.
<i>cardGuid</i>	The Card GUID part of the CardEase token. Note: This field, in combination with cardHash , replaces the need to provide the acctNumber field as a part of your enrollment check.	Length: 36 characters GUID
<i>cardHash</i>	The Card Hash part of the CardEase token. Note: This field, in combination with cardGuid , replaces the need to provide the acctNumber field as a part of your enrollment check.	
merchantId*	Acquirer-assigned Merchant identifier as boarded to the 3DS Server	Minimum Length: 0 Maximum Length: 35

Response

Field	Value	Format
enrolmentStatus	Whether or not PAN provided is enrolled in 3DS2	Length: 2 characters, numeric, left-padded with 0s 00 = Not enrolled with 3DS2 01 = Enrolled with 3DS2 02 - 09 = Reserved for future use.

threeDSMethod	Whether or not the Three DS Method is available.	Length: 1 character Y = Three DS Method is available N = Three DS Method is unavailable
---------------	--	---

Initialise Authentication

Path

/api/v2/auth/brw/init

Models

Request

Field	Value	Format
acctNumber*	Account number that will be used in the authorisation request for payment transactions.	Minimum Length: 13 Maximum Length: 19 Account number, numeric Mandatory when not using CardEase tokens.
<i>cardGuid</i>	The card GUID part of the CardEase token. Note: This field, in combination with cardHash , replaces the need to provide the acctNumber field as a part of your initialise authentication request.	Length: 36 characters GUID
<i>cardHash</i>	The card hash part of the CardEase token. Note: This field, in combination with cardGuid , replaces the need to provide the acctNumber field as a part of your initialise authentication request.	
eventCallbackUrl*	This is a 3DS requestor URL in which the 3DS requestor receives events during authentication. The 3DS requestor backend needs a handler method to accept the call from the 3DS Server. See the Event Callback section below for more details.	Min Length: 0 Max Length: 2000 Valid URL required
merchantId*	Acquirer-assigned Merchant identifier as boarded to the 3DS Server.	Minimum Length: 0 Maximum Length: 35

threeDSRequestorTransID*	Universal unique transaction identifier assigned by the 3DS Requestor to identify a single transaction. Must be unique per transaction.	Length: 36 characters GUID
---------------------------------	--	-----------------------------------

Response

Field	Value	Format
authUrl	<p>This URL is returned by the 3DS server that is to be used to perform authentication.</p> <p>The 3DS requestor makes a POST request to this URL when it receives the 3DSMethodSkipped or 3DSMethodFinished event as POSTed to the eventCallbackUrl to execute authentication.</p> <p>Please see the Authenticate section for more details on when this is used.</p>	<p>Min Length: 0 Max Length: 2000</p> <p>URL</p>
monUrl	<p>This URL returned by 3DS Server is used for the purpose of monitoring the timeout process for browser information collecting and 3DS method.</p> <p>It allows the 3DS Server to notify the 3DS requestor when the 3DS method times out.</p> <p>An iFrame should be created with the src attribute to allow for time outs within the Browser Fingerprinting process.</p> <p>Once the iFrame is created, it checks for the timeout for a maximum of 15 seconds (10 seconds for the 3DS method and 5 seconds for browser information collecting).</p> <p>The timer is started when the 3DS requestor calls the Intialise Authentication end-point and it will check periodically if the transaction with the given transaction ID has successfully called the Authenticate end-point.</p> <p>If a valid Authenticate request is not sent</p>	<p>Min Length: 0 Max Length: 2000</p> <p>URL</p>

	and the timer is up, it will notify the 3DS requestor through the iFrame to the eventCallbackUrl sent, with an event 'InitAuthTimedOut' so the 3DS requestor can terminate the transaction.	
threeDSServerCallbackUrl	This URL is returned by 3DS Server where the 3DS Requestor can use it to collect browser information as well as executing the 3DS method if supported by ACS for the BIN range.	Min Length: 0 Max Length: 2000 URL
threeDSServerTransactionID	Universal unique transaction identifier assigned by the 3DS Server to identify a single transaction.	Length: 36 characters GUID

Authentication

End-Point

Retrieved from the Initialise Authentication response field: **authUrl**

Models

Request

Top Level

Field	Value	Format
acctNumber*	Account number that will be used in the authorisation request for payment transactions.	Minimum Length: 13 Maximum Length: 19 Account number, numeric Mandatory when not using CardEase tokens.
cardExpiryDate*	Expiry Date of the PAN or token supplied to the 3DS Requestor by the Cardholder.	Length: 4 characters Format: YYYYMM Mandatory when not using CardEase tokens.
<i>cardGuid</i>	The card GUID part of the CardEase token. Note: This field, in combination with cardHash , replaces the need to provide the acctNumber and cardExpiryDate fields as a part of your authentication request.	Length: 36 characters GUID
<i>cardHash</i>	The card hash part of the CardEase token. Note: This field, in combination with cardGuid , replaces the need to provide the acctNumber and cardExpiryDate fields as a part of your authentication request.	
authenticationInd*	Indicates the type of Authentication request. This data element provides additional information to the ACS to determine the	Authentication Indicator

	best approach for handling an authentication request.	
browserInfo*	As provided by the param parameter as POSTed to the Event Callback page. Do not alter this field.	Minimum Length: 0 characters Maximum Length: 2000 characters
cardHolderInfo*	Nested JSON Field Note #1: The cardholderName part of this model is considered a mandatory field and must be supplied. Note #2: Please be aware of the difference in capitalisation of this field compared to the cardholderName field.	Card Holder Information
merchantId*	Acquirer-assigned Merchant identifier as boarded to 3DS Server.	Min Length: 0 characters Max Length: 36 characters
messageCategory*	This field must be specified to choose either payment authentication or non-payment authentication.	Payment Authentication: 01 or pa Non-payment Authentication: 02 or npa
threeDSServerTransactionID*	Universal unique transaction identifier assigned by the 3DS Server to identify a single transaction. Should be the same as provided in the Initialise Authentication response.	Length: 36 characters GUID
transType*	Identifies the type of transaction being authenticated. This field is mandatory for the Visa card scheme, hence the mandatory tagging. Additionally, this field is required in some markets, e.g. for Merchants in Brazil. Otherwise, it is optional.	Length: 2 characters See Appendix A for values.
acctID	Additional information about the account, optionally provided by the 3DS Requestor.	Maximum Length: 64 characters

acctInfo	<p>Additional information about the Cardholder's account provided by the 3DS Requestor.</p> <p>Optional, but strongly recommended to include to help reduce challenges.</p> <p>Nested JSON field.</p>	Account Information
acctType	<p>Indicates the type of account. For example, for a multi-account card product.</p> <p>Required if 3DS Requestor is asking Cardholder which Account Type they are using before making the purchase.</p> <p>This is also required in some markets (for example, for Merchants in Brazil).</p> <p>Otherwise, it is optional.</p>	Account Type
authenticationInfo	<p>Information about how the 3DS Requestor authenticated the cardholder before or during the transaction</p> <p>Optional, recommended to include</p> <p>Nested JSON Field</p>	Authentication Information
cardExpiryDate	<p>Expiry Date of the PAN or token supplied to the 3DS Requestor by the Cardholder.</p>	<p>Length: 4 characters</p> <p>Format: YYMM</p>
challengeInd	<p>Indicates whether a challenge is requested for this transaction.</p> <p>This field is optional.</p> <p>With messageCategory set to PA, a 3DS Requestor may have concerns about the transaction, and request a challenge.</p> <p>With messageCategory set to NPA, a challenge may be necessary when adding a new card to a wallet.</p>	<p>Length: 2 characters, 0 left-padded</p> <p>See Appendix A for values.</p>

merchantName	<p>Providing this field overrides the merchant name sent in the AReq.</p> <p>If this field is not provided, then the merchant name used when boarding the merchant is used.</p> <p>Can be used for scenarios where card schemes require the merchant name to be dynamic, such as the travel industry.</p> <p>Same name used in the authorisation message as defined in ISO 8583.</p>	<p>Min Length: 0 characters Max Length: 40 characters</p>
merchantRiskIndicator	Nested JSON field	Merchant Risk Indicator
payTokenInd	<p>EMV Payment Token Indicator</p> <p>A value of True indicates that the transaction was de-tokenised prior to being received by the ACS.</p> <p>This data element will be populated by the system residing in the 3-D Secure domain where the de-tokenisation occurs (i.e., the 3DS Server or the DS).</p> <p>Note: The Boolean value of true is the only valid response for this field when it is present.</p>	<p>Boolean</p> <p>true false</p>
priorTransID	<p>The 3DS Server Transaction ID for a prior authenticated transaction of a cardholder.</p> <p>It will fill the threeDSPriorAuthenticationInfo field of the AReq from the authenticationInfo field given in the previous transaction for the cardholder.</p> <p>Optional, recommended to include.</p>	<p>Length: 36 characters</p> <p>GUID</p>
purchaseAmount	<p>Purchase amount in minor units of currency with all punctuation removed.</p> <p>The purchaseCurrency will be used to determine the purchase currency exponent for the purchase amount if applicable.</p>	<p>Max Length: 48 characters</p> <p>Minor Units</p> <p>e.g. if specifying £123.45 you would provide the value 12345.</p>

	Required for PA or NPA (when 3DS Requestor Authentication Indicator = 02 or 03) transactions, otherwise it is Optional.	
purchaseCurrency	<p>Currency in which the purchase amount is expressed.</p> <p>If this value is not presented for PA or NPA (when 3DS Requestor Authentication Indicator = 02 or 03), 3DS Server will use the Default Currency as boarded to 3DS Server.</p> <p>If this value is present in the API request, it will be used instead of Default Currency boarded alongside the merchant.</p> <p>The purchaseCurrency will be used to determine the purchase currency exponent for the purchase amount if applicable.</p>	<p>Length: 3 characters</p> <p>Value should be an ISO 4217 three-digit currency code, other than those listed in Table A.5 in EMVCo Protocol and Core Functions Specification.</p>
purchaseDate	<p>Date and time of the purchase, expressed in UTC</p> <p>This field is always required for 01-PA and conditionally required for 02-NPA when 3DS Requestor Authentication Indicator = 02 or 03.</p>	<p>Format: yyyyMMddHHmmss</p> <p>e.g. 20180122153045</p>
purchaseInstalData	Indicates the maximum number of authorisations permitted for instalment payments.	<p>Length: 3 characters, number padded with left 0s.</p> <p>Numeric, from 001 to 999.</p>
recurringExpiry	<p>Date after which no further authorisations shall be performed.</p> <p>This field is always required for 01-PA for recurring transactions and conditionally required for 02-NPA when 3DS Requestor Authentication Indicator = 02 or 03.</p>	<p>Format: yyyyMMdd</p> <p>e.g. 20180131</p>
recurringFrequency	<p>Indicates the minimum number of days between authorisations.</p> <p>This field is always required for 01-PA for recurring transactions and conditionally</p>	<p>Max Length: 4</p> <p>Example Scenario</p> <p>If recurring transaction occurs 3,</p>

	required for 02-NPA when 3DS Requestor Authentication Indicator = 02 or 03.	5 and 9 days after the purchase date, then this field should be set to 2 because the minimum days is between after 3 and 5 days
threeDSRequestorTransID	<p>Universal unique transaction identifier assigned by the 3DS Requestor to identify a single transaction.</p> <p>Should be the same as provided in the Initialise Authentication request.</p>	<p>Length: 36 characters</p> <p>GUID</p>
transType	<p>Identifies the type of transaction being authenticated.</p> <p>This field is required in some markets, e.g. for Merchants in Brazil.</p> <p>Otherwise, it is optional.</p>	<p>Length: 2 characters</p> <p>See Appendix A for values.</p>

Account Information

Field	Value	Format
chAccAgeInd	Length of time that the cardholder has had the account with the 3DS Requestor.	Length: 2 characters, 0 left padded number 01 = No account (guest check-out) 02 = Created during this transaction 03 = Less than 30 days 04 = 30–60 days 05 = More than 60 days
chAccChange	Date that the cardholder's account with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment account, or new user(s) added.	Format: yyyyMMdd e.g. 20180131
chAccChangeInd	Length of time since the cardholder's account information with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment account, or new user(s) added.	Length: 2 characters, 0 left padded number 01 = Changed during this transaction 02 = Less than 30 days 03 = 30–60 days 04 = More than 60 days
chAccDate	Date that the cardholder opened the account with the 3DS Requestor.	Format: yyyyMMdd e.g. 20180131
chAccPwChange	Date that the cardholder's account with the 3DS Requestor had a password change or account reset.	Format: yyyyMMdd e.g. 20180131
chAccPwChangeInd	Indicates the length of time since the cardholder's account with the 3DS Requestor had a password change or account reset.	Length: 2 characters, 0 left padded number 01 = No change 02 = Changed during this transaction 03 = Less than 30 days 04 = 30–60 days 05 = More than 60 days

nbPurchaseAccount	Number of purchases with this cardholder account during the previous six months.	Maximum Length: 4 Min Value: 0 Max Value: 9999
paymentAccAge	Date that the payment account was enrolled in the cardholder's account with the 3DS Requestor.	Format: yyyyMMdd e.g. 20180131
paymentAccInd	Indicates the length of time that the payment account was enrolled in the cardholder's account with the 3DS Requestor.	Length: 2 characters, 0 left padded number 01 = No account (guest check-out) 02 = During this transaction 03 = Less than 30 days 04 = 30–60 days 05 = More than 60 days
provisionAttemptsDay	Number of Add Card attempts in the last 24 hours.	Maximum Length: 3 characters Min Value: 0 Max Value: 999
shipAddressUsage	Date when the shipping address used for this transaction was first used with the 3DS Requestor.	Format: yyyyMMdd e.g. 20180131
shipAddressUsageInd	Indicates when the shipping address used for this transaction was first used with the 3DS Requestor.	Length: 2 characters, 0 left padded number 01 = This transaction 02 = Less than 30 days 03 = 30–60 days 04 = More than 60 days
shipNameIndicator	Indicates if the Cardholder Name on the account is identical to the shipping Name used for this transaction.	Length: 2 characters, 0 left padded number 01 = Account Name identical to shipping Name 02 = Account Name different than shipping Name
suspiciousAccActivity	Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account.	Length: 2 characters, 0 left padded number 01 = No suspicious activity has been observed

		02 = Suspicious activity has been observed
txnActivityDay	Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours.	Max Length: 3 characters Min Value: 0 Max Value: 999
txnActivityYear	Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year.	Max Length: 3 characters Min Value: 0 Max Value: 999

Authentication Information

Field	Value	Format
threeDSReqAuthData	Data that documents and supports a specific authentication process.	Max Length: 2048 bytes JSON Data
threeDSReqAuthMethod	Mechanism used by the Cardholder to authenticate to the 3DS Requestor.	Length: 2 characters, 0 left padded number See Appendix A for values.
threeDSReqAuthTimestamp	Date and time in UTC of the cardholder authentication.	Format: YYYYMMDDHHMM e.g. 201711071307

Card Holder Information

Field	Value	Format
cardholderName*	<p>Name of the Cardholder</p> <p>Required unless market or regional mandate restricts sending this information.</p> <p>Note: Please be aware of the difference in capitalisation of this field compared to the cardHolderInfo field.</p>	<p>Min Length: 2 characters Max Length: 45 characters</p> <p>Note: Special characters (eg öüäéèê) in cardholder name are not allowed.</p> <p>Any non-ASCII characters should be transliterated to a close latin character.</p> <p>For example, any cases of the character “é”, should be replaced with “e” and “ø” should become “o”.</p>
billAddrCity	<p>The city of the Cardholder billing address associated with the card used for this purchase.</p> <p>For a pa authentication: required unless market or regional mandate restricts sending this information.</p> <p>For a npa authentication: Required (if available) unless market or regional mandate restricts sending this information.</p>	<p>Min Length: 0 characters Max Length: 50 characters</p>
billAddrCountry	<p>The country of the Cardholder's billing address associated with the card used for this purchase.</p> <p>pa: Required unless market or regional mandate restricts sending this information</p> <p>npa: Required (if available) unless market or regional mandate restricts sending this information.</p> <p>Required if the Cardholder Billing Address State is present.</p>	<p>Length: 3 characters</p> <p>Valid ISO 3166-1 three-digit country code, other than those listed in Table A.5. of EMVCo Protocol and Core Functions Specification.</p>
billAddrLine1	<p>First line of the street address or equivalent local portion of the Cardholder billing</p>	<p>Min Length: 0 characters Max Length: 50 characters</p>

	<p>address associated with the card used for this purchase.</p> <p>pa: Required unless market or regional mandate restricts sending this information.</p> <p>npa: Required (if available) unless market or regional mandate restricts sending this information</p>	
billAddrLine2	<p>Second line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.</p> <p>pa: Required unless market or regional mandate restricts sending this information.</p> <p>npa: Required (if available) unless market or regional mandate restricts sending this information.</p>	<p>Min Length: 0 characters</p> <p>Max Length: 50 characters</p>
billAddrLine3	<p>Third line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.</p> <p>pa: Required unless market or regional mandate restricts sending this information.</p> <p>npa: Required (if available) unless market or regional mandate restricts sending this information.</p>	<p>Min Length: 0 characters</p> <p>Max Length: 50 characters</p>
billAddrPostCode	<p>ZIP or other postal code of the Cardholder billing address associated with the card used for this purchase.</p> <p>pa: Required unless market or regional mandate restricts sending this information.</p> <p>npa: Required (if available) unless market or regional mandate restricts sending this information.</p>	<p>Min Length: 0 characters</p> <p>Max Length: 16 characters</p>
billAddrState	<p>The state or province of the Cardholder billing address associated with the card used for this purchase.</p>	<p>Min Length: 0 characters</p> <p>Max Length: 3 characters</p>

	<p>pa: Required unless market or regional mandate restricts sending this information.</p> <p>npa: Required (if available) unless market or regional mandate restricts sending this information.</p>	
email	<p>The email address associated with the account that is either entered by the Cardholder, or is on file with the 3DS Requestor.</p> <p>Required unless market or regional mandate restricts sending this information.</p>	<p>Min Length: 0 characters Max Length: 254 characters</p> <p>Must meet requirements of Section 3.4 of IETF RFC 5322.</p>
homePhone	<p>The home phone number provided by the Cardholder.</p> <p>Nested JSON field</p>	Phone Number
mobilePhone	<p>The mobile phone number provided by the Cardholder.</p> <p>Nested JSON field</p>	Phone Number
shipAddrCity	City portion of the shipping address requested by the Cardholder.	<p>Min Length: 0 characters Max Length: 50 characters</p>
shipAddrCountry	<p>Country of the shipping address requested by the Cardholder.</p> <p>Required if the Cardholder Shipping Address State is present.</p>	<p>Length: 3 characters</p> <p>Valid ISO 3166-1 three-digit country code, other than those listed in Table A.5. of EMVCo Protocol and Core Functions Specification.</p>
shipAddrLine1	First line of the street address or equivalent local portion of the shipping address requested by the Cardholder.	<p>Min Length: 0 characters Max Length: 50 characters</p>
shipAddrLine2	The second line of the street address or equivalent local portion of the shipping address requested by the Cardholder.	<p>Min Length: 0 characters Max Length: 50 characters</p>
shipAddrLine3	The third line of the street address or equivalent local portion of the shipping address requested by the Cardholder.	<p>Min Length: 0 characters Max Length: 50 characters</p>

shipAddrPostCode	The ZIP or other postal code of the shipping address requested by the Cardholder.	Min Length: 0 characters Max Length: 16 characters
shipAddrState	The state or province of the shipping address associated with the card being used for this purchase.	Min Length: 0 characters Max Length: 3 characters
workPhone	The work phone number provided by the Cardholder.	Phone Number

Phone Number

Field	Value	Format
cc	Country Code	Min Length: 1 characters Max Length: 3 characters
subscriber	Subscriber Sections of the Number	Min Length: 0 characters Max Length: 15 characters

Merchant Risk Indicator

Field	Value	Format
deliveryEmailAddress	For Electronic delivery, the email address to which the merchandise was delivered.	Min Length: 0 characters Max Length: 254 characters Email Address
deliveryTimeframe	Indicates the merchandise delivery timeframe.	Length: 2 characters 01 = Electronic Delivery 02 = Same day shipping 03 = Overnight shipping 04 = Two-day or more shipping
giftCardAmount	For prepaid or gift card purchase, the purchase amount total of prepaid or gift card(s) in major units	Min Length: 0 characters Max Length: 15 characters Minor Units e.g. if specifying £123.45 you would provide the value 12345.
giftCardCount	For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased.	Length: 2 characters Min Value: 0 Max Value: 99
giftCardCurr	For prepaid or gift card purchase, the currency code of the card.	Length: 3 characters Value should be an ISO 4217 three-digit currency code, other than those listed in Table A.5 in EMVCo Protocol and Core Functions Specification.
preOrderDate	For a pre-ordered purchase, the expected date that the merchandise will be available.	Length: 8 characters Format: YYYYMMDD e.g. 20180102
preOrderPurchaseInd	Indicates whether the Cardholder is placing an order for merchandise with a future availability or release date.	Length: 2 01 = Merchandise available 02 = Future availability

reorderItemsInd	Indicates whether the cardholder is reordering previously purchased merchandise.	Length: 2 characters 01 = First time ordered 02 = Reordered
shipIndicator	<p>Indicates shipping method chosen for the transaction.</p> <p>Merchants must choose the Shipping Indicator code that most accurately describes the cardholder's specific transaction, not their general business.</p> <p>If one or more items are included in the sale, use the Shipping Indicator code for the physical goods, or if all digital goods, use the Shipping Indicator code that describes the most expensive item.</p>	<p>Length: 2 characters</p> <p>See Appendix A for values.</p>

Response

Field	Value	Format
acsChallengeMandated	Indication of whether a challenge is required for the transaction to be authorised due to local/regional mandates or other variables.	Length: 1 Y = Challenge is mandated N = Challenge is not mandated
acsReferenceNumber	ACS Reference Number. Unique identifier assigned by the EMVCo Secretariat upon testing and approval.	Min Length: 0 characters Max Length: 32 characters
acsTransID	Universal unique transaction identifier assigned by the ACS to identify a single transaction.	Length: 36 characters GUID
authenticationType	Indicates the type of authentication method the Issuer used to challenge the Cardholder. Only present in challenge flow authentications.	Length: 2 characters See Appendix A for values
authenticationValue	Payment System-specific value provided as part of the ACS registration for each supported DS. For a PA authentication, this is required if Transaction Status = Y or A. Omitted from the RReq message when sent as an abandonment notification. For an NPA authentication; this is Conditional based on DS rules.	Length: 28 characters This is the value used alongside an authorisation to prove that the user was authenticated.
cardholderInfo	Text provided by the ACS/Issuer to the Cardholder during a Frictionless or Decoupled transaction. If this field is populated this information is required to be conveyed to the cardholder by the merchant.	Length: Up to 128 characters e.g. "Additional authentication is needed for this transaction, please contact (Issuer Name) at xxx-xxx-xxxx."
challengeUrl	The challenge URL returned by 3DS Server if a challenge is required. Only present in challenge flow authentications.	Min Length: 0 characters Max Length: 2000 characters URL

dsReferenceNumber	DS Reference Number. EMVCo-assigned unique identifier to track approved DS.	Min Length: 0 characters Max Length: 32 characters
dsTransID	Universal unique transaction identifier assigned by the DS to identify a single transaction.	Length: 36 characters GUID
eci	Payment System-specific value provided by the ACS to indicate the results of the attempt to authenticate the Cardholder. Defined by the DS.	Length: 2 characters
messageVersion	<p>Protocol version identifier.</p> <p>This shall be the Protocol Version Number of the specification utilised by the system creating this message.</p> <p>The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.</p>	Min Length: 5 Max Length: 8 e.g. 2.1.0
threeDSServerTransID	Universal unique transaction identifier assigned by the 3DS Server to identify a single transaction.	Length: 36 characters GUID
transStatus	Indicates whether a transaction qualifies as an authenticated transaction.	Length: 1 character See Appendix A for values.
transStatusReason	<p>Provides information on why the Transaction Status field has the specified value.</p> <p>For a PA authentication, this value is present if the Transaction Status field = N, U, or R.</p> <p>For a NPA authentication, may be conditionally returned as defined by the DS.</p>	Length: 2 characters See Appendix A for values.

Retrieve Results (GET request)

Path

```
/api/v2/auth/brw/result?threeDSServerTransID=<GUID>
```

Example

```
/api/v2/auth/brw/result?threeDSServerTransID=9c696eaa-7dbe-4bb4-b6fa-6f035c40da6f
```

Models

Request

Field	Value	Format
threeDSServerTransID*	The 3DS Server assigned Transaction ID related to this authentication attempt	Length: 36 characters GUID

Response

Field	Value	Format
acsReferenceNumber	ACS Reference Number. Unique identifier assigned by the EMVCo Secretariat upon testing and approval.	Min Length: 0 Max Length: 32 Alphanumeric
acsTransID	Universal unique transaction identifier assigned by the ACS to identify a single transaction.	Length: 36 characters GUID
authenticationType	Indicates the type of authentication method the Issuer used to challenge the Cardholder. Only present in challenge flow authentications	Length: 2 characters Numerical, left-padded with 0s.
authenticationValue	Payment System-specific value provided as part of the ACS registration for each supported DS. Authentication Value may be used to provide proof of authentication. With a PA authentication - this value will be present for any transactions with a transStatus of Y or A. With an NPA authentication - this value may be present	Length: 28 characters This is the value used alongside an authorisation to prove that the user was authenticated.

	depending on the Directory Server / scheme rules	
dsReferenceNumber	DS Reference Number. EMVCo-assigned unique identifier to track approved DS.	Length: Up to 32 characters
dsTransID	Universal unique transaction identifier assigned by the DS to identify a single transaction.	Length: 36 characters GUID This is a value used alongside an authorisation.
eci	Payment System-specific value provided by the ACS to indicate the results of the attempt to authenticate the Cardholder.	Length: 2 characters This is a value used alongside an authorisation.
interactionCounter	Indicates the number of authentication cycles attempted by the Cardholder. Only present in challenge flow authentications.	Length: 2 characters
messageVersion	Protocol version identifier. This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.	Minimum Length: 5 Maximum Length: 8 Example: 2.1.0 for 3DS 2.1 or 2.2.0 for 3DS 2.2. This is a value used alongside an authorisation.
threeDSSTransID	Universal unique transaction identifier assigned by the 3DS Server to identify a single transaction.	Length: 36 characters GUID This is a value used alongside an authorisation.
transStatus	Indicates whether a transaction qualifies as an authenticated transaction.	Length: 1 character See Appendix A for values.
transStatusReason	Provides information on why the Transaction Status field has the specified value. For a PA authentication, this will be present if the Transaction Status field = N, U, or R.	Length: 2 characters See Appendix A for values.

	For an NPA authentication, it may be present depending on the scheme Directory Server rules.	
--	--	--

Event Callback

The 3DS Server will use the eventCallbackUrl provided in the Initialise Authentication request in order to notify the 3DS Requestor of the next action to be taken.

This is done via a POST request to the URL.

POST Parameters

Parameter	Potential Values
requestorTransId	Universal unique transaction identifier assigned by the 3DS Requestor to identify a single transaction. This value will be as provided in the Initialise Authentication request.
param	With the events: 3DSMethodFinished and 3DSMethodSkipped and InitAuthTimedOut: This param will contain the browserInfo to be included as a part of the Authenticate request. With the event: 3DSMethodHasError This param will contain the specific error that occurred during the 3DS Method call.
event	See the below events section for details about this parameter.

Events

There are 4 events that need to be handled - the event will be passed back in the above event parameter, and can be one of the below values.

3DSMethodFinished

Browser information collection by the 3DS Server and ACS has finished the 3DS method successfully and collected additional browser information. The 3DS Server is ready to perform authentication.

This event will be sent through the threeDSSEServerCallbackUrl iFrame.

3DSMethodSkipped

Browser information collection by the 3DS Server is finished but 3DS method is not supported by the ACS, so the optional browser information collection was skipped. This event signals that the 3DS Server is ready to perform authentication.

This event will be sent through the threeDSServerCallbackUrl iFrame.

3DSMethodHasError

If the 3DS Server receives the 3DS method notification after the timeout of 10 seconds from ACS, this event will be sent.

The 3DS requestor should log this event for further troubleshooting with the ACS on why the 3DS method was delayed.

This event is only sent if the 3DS requestor has implemented the "monUrl".

If "monUrl" is not implemented, 3DSMethodFinished will be sent instead.

The "param" attribute will be set to the error message.

This event will be sent through the threeDSServerCallbackUrl iFrame.

InitAuthTimedOut

3DS method (10 seconds timeout) and browser information (5 seconds timeout) collecting has reached the timeout of 15 seconds. ACS 3DS method may be temporary unavailable, and this event can be used as a fallback mechanism e.g. retry authentication.

You are recommended to proceed to 3DS2 authentication upon receiving this event.

This event will be sent through the monUrl iFrame.

AuthResultReady

Cardholder has completed the challenge flow with the ACS. Final authentication result is ready to be requested from the 3DS Server. The 3DS Server receives the final authentication result receipt (RReq/RRes) from the DS/ACS.

This event will be sent through the challengeUrl iFrame.

Callback Example (PHP)

```
<?php

$trans_id = isset($_REQUEST["requestorTransId"]) ?
html_encode($_REQUEST["requestorTransId"]) : "";
$event = isset($_REQUEST["event"]) ? html_encode($_REQUEST["event"]) : "";
$param = isset($_REQUEST["param"]) ? html_encode($_REQUEST["param"]) : "";

switch($event)
{
    case "3DSMethodFinished":
        $callback = "threads_callback_method_finished";
        break;

    case "3DSMethodSkipped":
        $callback = "threads_callback_method_skipped";
        break;

    // $param here contains the error
    case "3DSMethodHasError":
        // Log error for troubleshooting purposes
        break;

    case "AuthResultReady":
        $callback = "threads_callback_auth_result_ready";
        break;

    case "InitAuthTimedOut":
        $callback = "threads_callback_method_init_auth_timeout";
        break;

    default:
        $callback = "threads_callback_method_unknown";
        break;
}

if(!isset($callback))
{
    // If there is no callback set, exit from script
    // This is because the 3DSMethodHasError doesn't require an action to be taken from
    this callback
    exit(0);
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"/>
    <title>3DSecure 2.0 Authentication</title>
</head>
<body>
<form>
    <input type="hidden" id="notifyCallback" name="notifyCallback"
data-th-value="<?php echo $callback; ?>"/>
    <input type="hidden" id="transId" name="transId" data-th-value="<?php echo
$trans_id; ?>"/>
    <input type="hidden" id="param" name="param" data-th-value="<?php echo $param;
?>"/>
</form>
```

```
<script>
    let callbackFn = parent["<?php echo $callback; ?>"];

    if (typeof callbackFn === 'function') {
        callbackFn("<?php echo $trans_id ?>", "<?php echo $param ?>");
    }
</script>
</body>
</html>
```

API Error Response

Errors can be returned by any request and at any point in the process.

The following HTTP Error codes may be used to signify an error:

- 400 (Invalid Request)
- 401 (Unauthorised)
- 403 (Forbidden)
- 500 (Internal Server Error)

Associated with these error codes, you may also receive an error response which will give you more information about the error that has been encountered.

Error Response

Field	Value	Format
errorCode	<p>Code indicating the type of problem identified in the message.</p> <p>Error codes having length of 3 are error codes defined by the EMVCo core specifications.</p> <p>Error codes having length of 4 are custom error codes.</p>	<p>Min Length: 3 characters Max Length: 4 characters</p> <p>Please see Error Codes for list of potential values.</p>
errorComponent	Code indicating the 3-D Secure component that identified the error.	<p>Length: 1 character</p> <p>C - Client</p> <p>S - 3DS Server</p> <p>D - Directory Server</p>

		A - ACS
errorDescription	Text describing the problem identified in the message.	Min Length: 0 characters Max Length: 2048 characters
errorDetail	Additional detail regarding the problem identified in the message.	
errorMessageType	Identifies the Message Type that was identified as erroneous. If the validation fails for the request sent from the 3DS requestor, this value is set to empty or 'AReq'.	Length: 4 characters AReq ARes CReq CRes RReq RRes Erro
messageType	Always returns value Erro	Length: 4 Always Erro
messageVersion	Protocol version identifier. This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction.	Minimum Length: 5 Maximum Length: 8 Example: 2.1.0 for 3DS 2.1 or 2.2.0 for 3DS 2.2.
sdkTransID	Universal unique transaction identifier assigned by the 3DS SDK to identify a single transaction.	Length: 36 characters GUID
threeDSServerTransID	Universal unique transaction identifier assigned by the 3DS Server to identify a single transaction.	Length: 36 characters GUID

Appendix A: API Values

TransStatus

Value	Description
Y	Authentication/ Account Verification Successful
N	Not Authenticated /Account Not Verified; Transaction denied
U	Authentication/ Account Verification Could Not Be Performed; Technical or other problem, as indicated in ARes or RReq
A	Attempts Processing Performed; Not Authenticated/Verified, but a proof of attempted authentication/verification is provided
C	Challenge Required; Additional authentication is required using the challenge set up
R	Authentication/ Account Verification Rejected; Issuer is rejecting authentication/verification and request that authorisation not be attempted.

TransStatusReason

Value	Description
01	Card authentication failed
02	Unknown Device
03	Unsupported Device
04	Exceeds authentication frequency limit
05	Expired card
06	Invalid card number
07	Invalid transaction
08	No Card record
09	Security failure
10	Stolen card
11	Suspected fraud
12	Transaction not permitted to cardholder
13	Cardholder not enrolled in service
14	Transaction timed out at the ACS
15	Low confidence
16	Medium confidence

17	High confidence
18	Very High confidence
19	Exceeds ACS maximum challenges
20	Non-Payment transaction not supported
21	3RI transaction not supported
80 ~ 90	Reserved for DS use

Account Type

Value	Description
01	Not Applicable
02	Credit
03	Debit
80 ~ 99	Reserved for DS use

Authentication Indicator

Value	Description
01	Payment transaction
02	Recurring transaction
03	Instalment transaction
04	Add card
05	Maintain card
06	Cardholder verification as part of EMV token ID and V
80 ~ 99	Reserved for DS use

AuthenticationType

Value	Description
01	Static
02	Dynamic
03	OOB
80 ~ 99	Reserved for DS use

ChallengeInd

Value	Description
01	No preference
02	No challenge requested
03	Challenge requested: 3DS Requestor Preference
04	Challenge requested: Mandate
80 ~ 99	Reserved for DS use

TransType

Value	Description
01	Goods/ Service Purchase
03	Check Acceptance
10	Account Funding
11	Quasi-Cash Transaction
28	Prepaid Activation and Load

Note: Values derived from the 8583 ISO Standard.

ThreeDSReqAuthMethod

Value	Description
01	No 3DS Requestor authentication occurred (i.e. cardholder “logged in” as guest)
02	Login to the cardholder account at the 3DS Requestor system using 3DS Requestor’s own credentials
03	Login to the cardholder account at the 3DS Requestor system using federated ID
04	Login to the cardholder account at the 3DS Requestor system using issuer credentials
05	Login to the cardholder account at the 3DS Requestor system using third-party authentication
06	Login to the cardholder account at the 3DS Requestor system using FIDO Authenticator
80 ~ 99	Reserved for DS use

ShipIndicator

Value	Description
01	Ship to cardholder's billing address
02	Ship to another verified address on file with merchant
03	Ship to address that is different than the cardholder's billing address
04	"Ship to Store" / Pick-up at local store (Store address shall be populated in shipping address fields)
05	Digital goods (includes online services, electronic gift cards and redemption codes)
06	Travel and Event tickets, not shipped
07	Other (for example, Gaming, digital services not shipped, emedia subscriptions, etc.)

Error Codes

3DS Error Codes

Value	Name	HTTP Code	Description
101	MESSAGE_RECEIVED_INVALID	400	Received message is invalid. Message is not AReq, ARes, CReq, CRes, PReq, PRes, RReq, or RRes. For example, 3DS Server receives an message from DS as a response to AReq that is not ARes or Erro message.
102	MESSAGE_VERSION_NUMBER_NOT_SUPPORTED	400	Unsupported message version number. Message Version Number received is not valid for the receiving component. For example, DS sends a messageVersion field set to an invalid value, or value that is not supported by the ACS.
201	REQUIRED_DATA_ELEMENT_MISSING	400	A message element required as defined according to the specification is missing. This error code will be returned if any of the fields marked as required is missing in the request.
202	CRITICAL_MESSAGE_EXTENSION_NOT_RECOGNISED	400	Message extension that is critical is not present.
203	FORMAT_OF_ONE_OR_MORE_DATA_ELEMENTS_IS_INVALID_ACCORDING_TO_THE_SPECIFICATION	400	Data element is not in the required format or value is invalid as defined according to the specification.
204	DUPLICATE_DATA_ELEMENT	400	Found duplicate data elements in the request.
301	TRANSACTION_ID_NOT_RECOGNISED	400	Transaction ID received is not valid for the receiving component.
302	DATA_DECRYPTION_FAILURE	500	Data could not be decrypted by the receiving system due to technical or other reasons.
303	ACCESS_DENIED_INVALID_ENDPOINT	401	Endpoint for the API request is invalid. Check the requesting URL.

304	ISO_CODE_INVALID	400	ISO code not valid according to ISO tables (for either country or currency).
305	TRANSACTION_DATA_NOT_VALID	400	Transaction data is invalid. Please refer to the error description to find out why the transaction data was invalid.
306	MERCHANT_CATEGORY_CODE_MCC_NOT_VALID_FOR_PAYMENT_SYSTEM	400	Merchant category code is invalid. Invalid MCC received in the AReq message and DS may throw this error back to the 3DS Server.
402	TRANSACTION_TIMED_OUT	408	Transaction has timed out.
403	TRANSIENT_SYSTEM_FAILURE	500	System has failed for a short period. For example, a slowly processing back-end system.
404	PERMANENT_SYSTEM_FAILURE	500	System has failed permanently. For example, a critical database cannot be accessed.
405	SYSTEM_CONNECTION_FAILURE	500	Failed to connect to the system. For example, the sending component is unable to establish connection to the receiving component.

Transaction Error Codes

Value	Name	HTTP Code	Description
1000	DIRECTORY_SERVER_NOT_AVAILABLE	500 / 402	<p>If any errors occurred during the connection to Directory Server this error code may be returned.</p> <p>Error code 402 is returned instead if the reason for the connection error was because of timeout.</p>
1002	ERROR_SAVE_TRANSACTION	500	Error occurred while saving transaction.
1004	UNHANDLED_EXCEPTION	500	An unhandled exception occurred during the transaction. Please report this if encountered.
1013	INVALID_TRANSACTION_ID	400	Transaction ID of 3DS Server is not recognised.
1014	INVALID_REQUESTOR_TRANSACTION_ID	400	Transaction ID of 3DS Requestor is not recognised
1016	MISSING_REQUIRED_ELEMENT	400	Required element missing.
1020	ERROR_TRANSMISSION_DATA	500	Thrown when there are errors in data transmission between any two 3DS components.
1021	PRIOR_TRANS_ID_NOT_FOUND	400	Prior Transaction ID could not be found in the database, or is invalid.
1022	INVALID_FORMAT	400	Format of one or more elements is invalid according to the EMV Co specification.
1026	MERCHANT_ID_THREEDS_REQUESTOR_ID_INVALID	400	Invalid merchantId is given to the authentication request.

General Error Codes

Value	Name	HTTP Code	Description
2002	VALIDATION_ERROR	400	Invalid Inputs - may be returned if the request is not properly formatted JSON.
2005	ACCESS_DENIED	401	Access is denied - check the error detail for more information.
2007	INTERNAL_SERVER_ERROR	500	An internal server error has occurred - check the error detail for more information.
2009	SESSION_TIMED_OUT	408	Session has timed out. Can be returned if the transaction has already been finished and you re-attempt a request.

Appendix B: Glossary

Term	Acronym	Description
3DS Client		<p>The cardholder / consumer facing component of EMV 3-D Secure (3DS).</p> <p>In browser based EMV 3DS, this part facilitates interaction between the cardholder / consumer's browser and the 3DS Requestor.</p>
3DS Requestor		<p>The initiator of the EMV 3-D Secure Authentication Request, known as the AReq message.</p> <p>As an example, this may be a hosted payment page or in the case of a direct integrator, it will be the direct integrators website.</p>
3DS Server	3DSS	For the purposes of this documentation, this refers to the CardEase 3DS Server.
3-D Secure	3DS	<p>This stands for Three Domain Secure.</p> <p>This is a fraud prevention authentication protocol that enables the secure processing of eCommerce payment transactions, and helps ensure that the person performing the payment is entitled to make payments.</p>
Access Control Server	ACS	<p>This is the 3-D Secure component that resides in the issuer domain.</p> <p>This is the component that verifies whether authentication is available for a specific card number, and also facilitates authentication of specific cardholders / consumers.</p>
Authentication		The process of confirming that the person performing an eCommerce transaction has permission and is entitled to use the payment card before performing any authorisations.
Authentication Request Message	AReq	<p>One of the EMV 3-D Secure messages sent by the 3DS Server to initiate the authentication process.</p> <p>This goes via the DS to the ACS to initiate the EMV 3-D Secure authentication process.</p>
Authentication Response Message	ARes	<p>The response message to the AReq message.</p> <p>This is returned by the ACS via the DS.</p>
Authentication Value	AV	This is a cryptographic value generated by the ACS with the purpose of allowing the authorisation system to validate the integrity of the authentication result.

		<p>This will be required as a part of your authorisation.</p> <p>The algorithm used for this is defined by each Payment System.</p>
Challenge / Challenge Flow		<p>This is when the cardholder / consumer is required to authenticate themselves directly with the ACS.</p>
Challenge Request Message	CReq	<p>The request sent by the 3DS Server to the ACS in order to support the authentication process.</p> <p>This is where additional information about the cardholder is sent to the ACS in order to help facilitate the 3D Secure process.</p>
Challenge Response Message	CRes	<p>The response to the CReq message, sent by the ACS.</p> <p>This can indicate a successful authentication, or alternatively signal that further cardholder interaction is required in order to complete the authentication via a challenge.</p>
Consumer Device		<p>The cardholder device used to conduct authentications and payments.</p> <p>Some examples of a consumer device could be a laptop, or smartphone.</p>
Directory Server	DS	<p>This is a component that resides in the interoperability domain of the 3-D Secure specification.</p> <p>It performs a few different functions that includes authentication of the 3DS server, routing messages between the 3DS Server and the different ACS servers operated by issuers.</p> <p>These tend to be managed by the card schemes</p>
Electronic Commerce Indicator	ECI	<p>This value is provided by the ACS and indicates the results of the authentication attempt.</p> <p>This will be required as a part of your authorisation.</p>
Frictionless / Frictionless Flow		<p>This is when the cardholder / consumer isn't required to explicitly authenticate themselves with the ACS.</p> <p>As there is no cardholder / consumer interaction, this process can happen invisibly to the cardholder / consumer.</p>
Non-Payment Authentication	NPA	<p>EMV 3DS authentication type in which no transaction is attached. Used for identity verification.</p>
Results Request Message	RReq	<p>The request sent by the ACS (via the DS) that allows the results of an authentication to reach the 3DS Server.</p>

Results Response Message	RRes Y	<p>The response to the RReq message sent by the 3DS Server to the ACS (via the DS).</p> <p>This allows the 3DS Server to acknowledge receipt of the RReq message.</p>
--------------------------	-----------	---

Appendix C: Test Platform Details

Test Merchant

On our test platform, only one merchant may be used.

Merchant ID	Merchant Name	Merchant Token
123456789012345	Test Card	b1cdf956-a4f4-4ce4-ade6-cd84d68e59f2

The Merchant Token is the required value for the `C3DS-Merchant-Token` header.

The Merchant ID should be used with any API requests that have the field `merchantId` as a part of the request.

Test Cards

Below are the details of the various test card numbers that can be used to test the different flows and results that can come from an EMV 3DS 2.1 authentication.

Expiry Date / Cardholder Name

The expiry date and cardholder name for all these cards is always the same.

Important: These **must** be used otherwise any requests sent to the 3DS Server are likely to fail.

Expiry Date	Cardholder Name
08/2025	Test Card

Successful Frictionless Flow Authentication

Scheme	Card Number
Amex	3400000000000108
Discover/Diners	6440000000000104
Discover/Diners	360000000000008
Mastercard	5100000000000107
Visa	4100000000000100

3DS authentication will complete successfully without any challenge from the ACS.

ARes Result:

- Transaction Status = Y
- ECI = 05, or 02 (for Mastercard)
- Contains an Authentication Value

Successful Challenge Flow Authentication

Scheme	Card Number
Amex	340000000005008
Discover/Diners	644000000005004
Discover/Diners	36000000005007
Mastercard	510000000005007
Visa	410000000005000

These card numbers will trigger a step up to a challenge and request the user enter a password.

The password 123456 should be used in order to successfully authenticate the user.

ARes Result:

- Transaction Status = C

RReq Result:

- Transaction Status = Y
- ECI = 05, or 02 (for Mastercard)
- Contains an Authentication Value

Authentication Attempted

Scheme	Card Number
Amex	3400000000100007
Discover/Diners	64400000000100003
Discover/Diners	360000000100006
Mastercard	51000000000100006
Visa	41000000000100009

This transaction will attempt to perform authentication before returning an “Attempted” response.

ARes result:

- Transaction Status = A
- ECI = 06 or 01 (Mastercard)
- Contains an Authentication Value

Authentication Failed

Scheme	Card Number
Amex	340000000300003
Discover/Diners	6440000000300009
Discover/Diners	36000000300002
Mastercard	5100000000300002
Visa	4100000000300005

With these test PANs, a Challenge step up will be initiated and the user asked for a password.

The password 111111 should be used in order to successfully trigger the failure case.

ARes Result:

- Transaction Status = C

RReq Result:

- Transaction Status = N
- ECI = 00
- Does not contain an Authentication Value

Authentication Unavailable

Scheme	Card Number
Amex	3400000000400001
Discover/Diners	6440000000400007
Discover/Diners	360000000400000
Mastercard	5100000000400000
Visa	4100000000400003

Transaction will end with the authentication being rejected by the ACS.

ARes result:

- Transaction Status = U
- ECI is dependant on card scheme specific rules
- Does not contain an Authentication Value

Authentication Rejected

Scheme	Card Number
Amex	340000000500008
Discover/Diners	6440000000500004
Discover/Diners	36000000500007
Mastercard	5100000000500007
Visa	4100000000500000

Transaction will end with the authentication being rejected by the ACS.

ARes result:

- Transaction Status = R
- ECI is dependant on card scheme specific rules
- Does not contain an Authentication Value