SmartVista 3DS Server Integration

Developer Reference

PRODUCT DOCUMENTATION

14 October 2020

bpcbt.com

Notice

For information regarding permissions, write to:

BPC Banking Technologies
Neuhofstrasse 5a
6340 Baar
Switzerland

Phone: +41 43 508 4024

Email: info@bpcbt.com

Web: www.bpcbt.com

File: sv_3ds-server-1.32_dev-ref__en_core_20201014

Applicable to: SmartVista 3DS Server 1.32

Permitted distribution: Under NDA only

Contents

Scope

SmartVista 3DS Server is a standalone component that can be integrated with an acquirer's e-commerce payment solution.

The acquirer e-commerce payment solution is further referred to as SmartVista E-commerce Payment Gateway (EPG).

The current version of the document describes the settings applicable to 3-D Secure version 2.1.0.

Version history

Date	Contributors	Summary of changes
2019-06-27	A.G.	Initial version of the document.
2020-04-01	E.Z.	The **`returnUrl`** parameter has been added to the **`sendAReq.do`** description (in Data transfer for Authentication Request Message). The Swagger UI section has been added.
2020-07-30	E.Z.	API method names have been changed.
2020-10-14	E.Z.	The Validating the application operation section has been added. The API method description has been updated and the following methods have been added: • Request for the authentication result notification • Request for RReq by transaction ID • RReq request for RRes • Browser data collection request • Browser data transfer request • Error data collection request Appendix 1 has been added with request and response examples.

1 Transaction flow

The transaction flow among the e-commerce infrastructure components is depicted in the figure below.

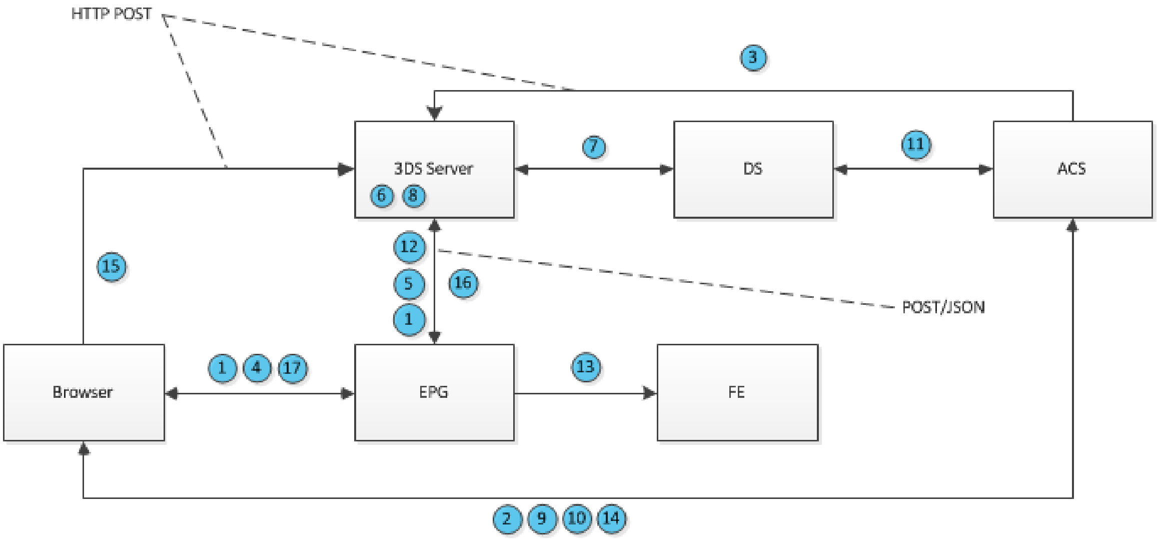


Figure 1. 3DS 2 transaction flow

1. Browser via EPG calls the method to check PAN ranges for participation in 3DS 2 technology. The response contains **`threeDsServerTransactionId`**, **`acsMethodUrl`** and a packaged data block with 3dsMethodData and URL (for the 3DS Server to collect browser data).
2. Browser opens iFrame and sends 3dsMethodData to acsMethodUrl, opens the second frame for data collection by 3DS Server.
3. ACS sends notification about 3dsMethod completion.
4. Browser sends payment request to EPG.
5. EPG sends the data to the 3DS Server to build an Authentication Request.
6. The 3DS Server waits for a notification on the 3dsMethod completion and then builds AReq.
7. The 3DS Server sends AReq to ACS through DS.
8. The 3DS Server receives ARes from ACS trough DS and, if needed, builds CReq and returns it.
9. Browser sends CReq to ACS.
10. OTP is entered by the cardholder via browser and submitted to ACS.
11. ACS sends RReq to the 3DS Server through DS.
12. The 3DS Server calls respective EPG API and submits results from RReq.
13. EPG sends financial authorization request.
14. ACS sends CRes to browser.
15. Browser forwards CRes to 3DS Server.
16. The 3DS Server redirects browser to EPG including CRes in the data.

In case of Frictionless authentication, the process will end at AReq/ARes stage. Data for financial authorization request will be sent in the Authentication Response.

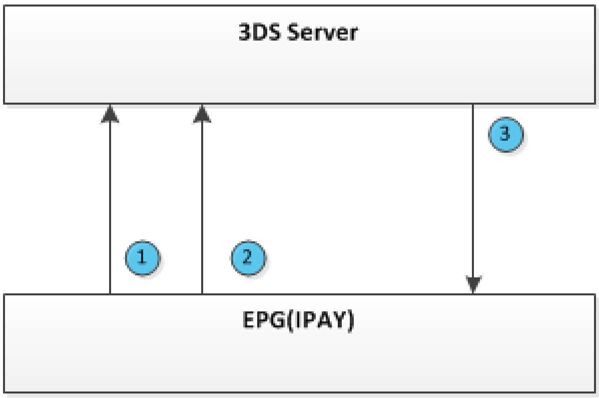
2 Integration

To integrate the acquirer's system with the 3DS Server, you must add the following artifact to the project dependencies:

```
<dependency>

<groupId>com.bpcbt.ecom</groupId>

<artifactId>ecom-3ds2-components-api</artifactId>

<version>1.1.0</version>

</dependency>
```

A controller must be created in the acquirer's e-commerce payment solution according to the **com.bpcbt.ecom.threeds2.api.Requestor** interface and the 3DS Server must be called according to the **com.bpcbt.ecom.threeds2.api.Server** interface.

Methods on both sides accept only POST requests with a corresponding JSON body.



1. EPG calls the **/api/v1/card** service and passes the card number (**PAN**) and **deviceChannel**, and, optionally, **acquirerBIN** and **threeDSRequestorID**. The 3DS Server checks if PAN falls into the PAN ranges stored the in the PRes cache and returns **CardCheckResponse**. EPG opens two iFrames on the checkout page:
   - In the first iFrame it sends **threeDSMethodDataPacked** by a POST request to threeDSMethodURL.
   - In the second iFrame it sends a request to **threeDSMethodURLServer**.
2. EPG calls the **/api/v1/auth** service. The 3DS Server extends the AReq (adds the fields **returnURL** (not used) and **dsEmulator** used if the interaction with DS is unnecessary) and sends it to DS. If the cardholder interaction is required (challenge authentication), the 3DS Server returns a packed CReq to be sent to ACS.
3. To get the authentication result, the 3DS Server calls the URL that is specified in the `application-prod.yml` file as the **config.acquirers.callbackResultUrl** parameter value (see *3DS Server Installation and Configuration Guide*). The callback URL defines the

link where to send a notification to the 3DS Requestor about the authentication completion (currently it is a POST request to https://epg.bpcbt.com/api/v1/3ds2/result).

3 Authentication between 3DS Server and other components

This section describes the 3DS Server authentication settings used in communication with external systems.

3.1 Payment gateway (EPG) authentication

If 3DS Server is integrated with an external payment gateway (EPG), each call between the 3DS Server and the payment gateway (EPG) must contain an authorization token. The external system (payment gateway) that initiates a call generates a JSON Web Token (JWT) token using a shared AES 256 key. The algorithm for the token signing is HMAC-SHA.

The JWT settings for communication with an external system are contained in the `application-prod.yml` file of the 3DS Server package in the **`jwt`** block (see *3DS Server Installation and Configuration Guide*):

```
jwt:
  enabled: true
  # key alias for JWT from config.identitystorePath
  keyAlias: local
  # JWT can independently verify it and check whether it has expired
  expirationTimeout: 60_000
  # |shifSeconds| > 2 seconds only for testing purposes
  shiftSeconds: -2
  # JWT Identity keystore type
  identitystoreType: jks
  # Path to JWT Identity keystore which holds JWT keys
  identitystorePath: ./jks/jwt-keystore.jks
  # JWT Identity keystore password
  identitystorePassword: ENC(...)
```

If **`jwt.enabled`** is set to `true`, the **HEADER** value `Authorization` is added to each request (api/v1/card or api/v1/auth).

The address of the **`JwtService`** service is `com.bpcbt.ecom.threeds2.security.JwtService`.

The token **`Subject`** attribute should be set to “**`3ds2_integration`**” without quotes. The token attributes “`Not before`” and “`Expiration`” should be set according to the requirements. The token itself should be placed in the “`Authorization`” header of HTTP request.

The following conditions must be met for authentication with the 3DS Server:

- The **`com.bpcbt.ecom.threeds2.security.JwtService`** API must be used (from **`ecom-3ds2-components-api`**).
- A 256-bit AES key must be passed to the constructor. The same key must be stored in the 3DS Server keystore.
- All calls and requests to the 3DS Server must contain the `JwtService.HEADER_STRING` header. This header must contain a token. The 3DS Server will also add this header to its calls or requests. The header must be verified.

3.2 DS authentication

Mutual authentication must be configured for the connection with DS.

3.3 Certificates

The following certificate types are used within the 3DS 2 infrastructure:

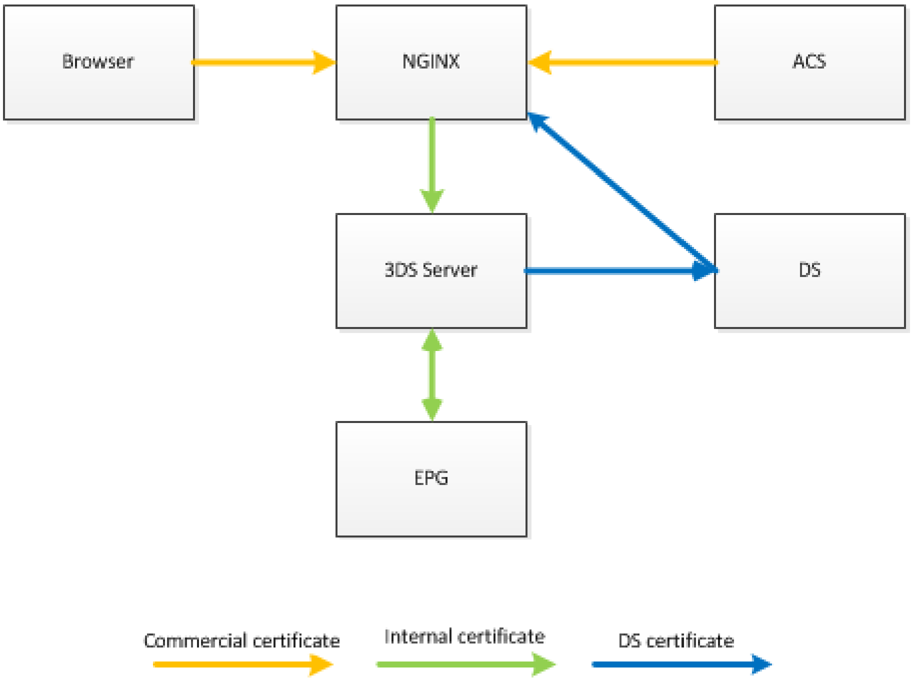


Figure 2. Certificate types

4 Validating the application operation

To ensure that the 3DS Server application is running and operates properly, execute any ping command (GET, HEAD, POST, and so on) for any context, for example:

```
$ curl -v http://<3DS Server URL>:<Port>/ping
```

The application returns a line with a **pong** when working properly.

4.1 Checking the application version

You can also check the successful installation of the application by sending a request to check its version. To check the application version, send any request with **v** at the end of the line, for example:

```
curl -v -XPOST http://<3DS Server URL>:<Port>/v
```

The application will return the recommended version from the versions supported by the application (specified in the **`messageType`** parameter).

5 API

The API methods used for the integration with the 3DS Server are described below.

You can find the API methods in the Swagger UI that is available at the following address: `http://`*<3DS Server URL>*`/swagger-ui.html`.

5.1 3-D Secure participation eligibility

To check is a card is eligible for 3-D Secure operations (figure 1, step 1), the **`/api/v1/card`** method is used.

> **Note:** There are equivalent methods with different names. Those methods are obsolete, use the method mentioned above.

The POST method is used to send the request.

Request parameters

Name	Type	Mandatory	Description
pan	N..20	Yes	Primary Account Number.
deviceChannel	string	Yes	Indicates the type of channel interface that is used to initiate the transaction Available value ranges: • 01–03, • 80–99
acquirerBIN	N..11	No	Acquirer bank identification number. An optional criterion to search for a card range.

Name	Type	Mandatory	Description
threeDSRequestorID	N..358	No	**Note:** This parameter has not been implemented yet. Identifier of the 3DS Requestor assigned by DS. Each DS provides a unique ID to each 3DS Requestor individually.

Response

An HTTP status (with a code) is returned as a response. In the case of a success status (`200`), it is returned with a JSON structure of parameters detailing the applicable 3-D Secure checks.

Name	Type	Mandatory	Description

**HTTP/*<version>*
*<status_code>***
	N3	No	Response code that reflects the status:
			• 200 – OK
			This code is returned with a JSON structure. See the structure description below. Example:
			`{`
			`  "is3Ds2Eligible": true,`
			`  "protocolVersion":`
			`"string",`
			`  "threeDSMethodDataPacked":`
			`"string",`
			`  "threeDSMethodURL":`
			`"string",`
			`  "threeDSMethodURLServer":`
			`"string",`
			`  "threeDSServerTransID":`
			`"string"`
			`}`
			• 400 – Wrong deviceChannel
			• 405 – Wrong method is used to send the request

Name	Type	Mandatory	Description
			- 501 – threeDSRequestorID has been transferred that is not implemented

The response structure contents

The following parameters are transferred in a JSON structure:

Name	Type	Description
is3Ds2Eligible	B	Indicates whether card is eligible for 3DS2.
threeDSServerTransID	AN36	Unique 3DS Server transaction identifier Must be passed in the subsequent authentication request
protocolVersion	ANS	Supposedly may indicate existing 1.0.2 version May be removed in future.
threeDSMethodDataPacked	ANS..512	Base64 packed data block to send to ACS **`threeDSMethodURL`**.
threeDSMethodURL	ANS..256	ACS URL to collect cardholder browser data
threeDSMethodURLServer	ANS..256	3DS Server URL to collect cardholder browser data. 3DS Requestor should open iframe and send POST request with **`threeDSServerTransID`** to this URL

5.2 AReq request

To send the authentication request with payment data to 3DS Server for AReq (figure 1, step 5), the **`/api/v1/auth`** method is used.

> **Note:** There are equivalent methods with different names (such as **`/rest/server/auth`** or **`/rest/server/sendAReq`**). Those methods are obsolete, use the method mentioned above.

The POST method is used to send the request.

Request body

The request is sent as a body with the AReq (authentication request) contents. See an example of the value in Appendix 1: Request and response examples.

The request body contents:

The content type is `application/json`.

Name	Type	Mandatory	Description
messageVersion	ANS..8	Yes	Protocol version identifier. The Message Version Number is set by the 3DS Server.
messageCategory	N2	Yes	Message Category. Identifies the category of the message for a specific use case. Values accepted: • `01 – Payment Authentication` • `02 - Non-Payment Authentication`
messageType	A4	Yes	Message Type. Value accepted: • `AReq`
returnUrl	AN..512	Yes	URL to which the customer is redirected after a successful payment.
threeDSServerTransID	AN36	Yes	Unique 3DS Server transaction identifier. It must be passed in a subsequent authentication request.
threeDSRequestorAuthenticationInd	N2	Yes	3DS Requestor Authentication Indicator. Indicates the type of Authentication request. Value accepted: • `01 = Payment transaction;` • `02 = Recurring transaction;` • `03 = Instalment transaction;` • `04 = Add card;` • `05 = Maintain card;`

Name	Type	Mandatory	Description
			• `06` = Cardholder verification as part of EMV token ID&V.
threeDSRequestorAuthenticationInfo		No	Object with the parameters described below:
threeDSReqAuthData	ANS..2048	Yes	3DS Requestor Authentication Data. Data that documents and supports a specific authentication process. For each 3DS Requestor Authentication Method, this field carries data that the ACS can use to verify the authentication process.
threeDSReqAuthMethod	N2	Yes	3DS Requestor Authentication Method. Mechanism used by the Cardholder to authenticate to the 3DS Requestor. Value accepted: • `01` = No 3DS Requestor authentication occurred; • `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.

Name	Type	Mandatory	Description
threeDSRequestorChallengeInd	N2	Yes	3DS Requestor Challenge Indicator. Indicates whether a challenge is requested for this transaction. Value accepted: • `01` = No preference; • `02` = No challenge requested; • `03` = Challenge requested: 3DS Requestor Preference; • `04` = Challenge requested: Mandate.
threeDSRequestorURL	ANS..256	Yes	Fully qualified URL of 3DS Requestor service. This service will be called by the 3DS Server upon receiving RReq from DS. In this implementation it is the **`processResult.do`** service.
notificationURL	ANS..256	Yes	Fully qualified URL of the service that receives the CRes message or Error Message.
acquirerBIN	AN11	Yes	Acquiring institution identification code defined by each Payment System or DS.
acquirerMerchantID	AN35	Yes	Acquirer-assigned Merchant identifier. This may be the same value that is used in authorization requests sent on behalf of the 3DS Requestor and is represented in ISO 8583 formatting requirements.
merchantName	ANS..40	Yes	Merchant name assigned by the Acquirer or Payment System.
merchantCountryCode	AN3	Yes	Country Code of the Merchant.
mcc	N4	Yes	Merchant Category Code.

Name	Type	Mandatory	Description
acctNumber	N..19	Yes	Cardholder Account Number.
cardExpiryDate	N4	Yes	Card/Token Expiry Date. The date format is *yyMM*.
purchaseAmount	N..48	Yes	Purchase amount in minor units of currency with all punctuation removed.
purchaseCurrency	N3	Yes	Currency of the purchase amount.
purchaseExponent	N1	Yes	Minor units of currency as specified in the ISO 4217 currency exponent.
purchaseDate	N14	Yes	Date and time of the purchase, in the UTC format. The date format is *yyyyMMddHHmmss*.
cardholderName	ANS2..45	Yes	Name of the Cardholder.
deviceChannel	AN..128	Yes	Indicates the type of channel interface being used to initiate the transaction. Values accepted: • APP_BASED • BROWSER • THREDS_REQUESTOR_INITIATED
For App-based implementation (deviceChannel = APP_BASED)			
sdkAppID	ANS36	Yes	SDK App ID. Universally unique ID created upon all installations and updates of the 3DS Requestor App on a Consumer Device.
sdkEphemPubKey	ANS..256	Yes	SDK Ephemeral Public Key. Public key component of the ephemeral key pair generated by the 3DS SDK and used to

Name	Type	Mandatory	Description
			establish session keys between the 3DS SDK and ACS.
sdkMaxTimeout	N2	Yes	SDK Maximum Timeout. Indicates maximum amount of time (in minutes) for all exchanges.
sdkReferenceNumber	ANS..32	Yes	SDK Reference Number. Identifies the vendor and version for the 3DS SDK that is integrated in a 3DS Requestor App, assigned by EMVCo when the 3DS SDK is approved.
sdkTransID	ANS36	Yes	SDK Transaction ID. Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction.
acctInfo			Cardholder Account Information. Additional information about the Cardholder's account provided by the 3DS Requestor.
chAccAgeInd	N2	No	Cardholder Account Age Indicator. Length of time that the cardholder has had the account with the 3DS requestor. Value accepted: • `01 = No account;` • `02 = Created during this transaction;` • `03 = Less than 30 days;` • `04 = 30-60 days;` • `05 = More than 60 days.`
chAccChange	N8	No	Cardholder Account Change. Date that the cardholder's account with the 3DS

Name	Type	Mandatory	Description
			Requestor was last changed, including Billing or Shipping address, new payment account, or new user(s) added. The date format is *yyyyMMdd*.
chAccChangeInd	N2	No	Cardholder Account Change Indicator. 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. Value accepted: • `01 = No change;` • `02 = Changed during this transaction;` • `03 = Less than 30 days;` • `04 = 30-60 days;` • `05 = More than 60 days.`
chAccDate	N8	No	Cardholder Account Date. Date that the cardholder opened the account with the 3DS Requestor. The date format is *yyyyMMdd*.
chAccPwChange	N8	No	Cardholder Account Password Change. Date that cardholder's account with the 3DS Requestor had a password change or account reset. The date format is *yyyyMMdd*.
chAccPwChangeInd	N2	No	Cardholder Account Password Change Indicator. Indicates the length of time since the cardholder's account with the 3DS Requestor had a password change or account reset. Value accepted: • `01 = No account;` • `02 = Created during this transaction;`

Name	Type	Mandatory	Description
			- `03 = Less than 30 days;` - `04 = 30-60 days;` - `05 = More than 60 days.`
shipAddressUsage	N8	No	Shipping Address Usage. Date when the shipping address for this transaction was first used with the 3DS Requestor. The date format is *yyyyMMdd*.
shipAddressUsageInd	N2	No	Shipping Address Usage Indicator. Indicates when the shipping address used for this transaction was first used with the 3DS Requestor.
shipNameIndicator	N2	No	Shipping Name Indicator. Indicates if the Cardholder Name on the account is identical to the shipping Name used for this transaction.
suspiciousAccActivity	N2	No	Suspicious Account Activity. Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account.
email	ANS..254	Yes	Cardholder Email Address. The email address associated with the account that is either entered by the Cardholder, or is on file with the 3DS Requestor.
homePhone		Yes (if available)	Home phone number provided by the Cardholder.
cc	N..3	Yes	Country Code.

Name	Type	Mandatory	Description
subscriber	N..15	Yes	Subscriber
addrMatch	B	No	Address Match Indicator. Indicates whether the Cardholder Shipping Address and Cardholder Billing Address are the same.
billAddrCity	ANS..50	No	Cardholder Billing Address City. The city of the Cardholder billing address associated with the card used for this purchase.
billAddrCountry	N3	No	Cardholder Billing Address Country. The country of the Cardholder billing address associated with the card used for this purchase.
billAddrLine1	ANS..50	No	Cardholder Billing Address Line 1. First line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.
billAddrLine2	ANS..50	No	Cardholder Billing Address Line 2. Second line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.
billAddrLine3	ANS..50	No	Cardholder Billing Address Line 3. Third line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.
billAddrPostCode	AN..16	No	Cardholder Billing Address Postal Code. ZIP or other postal code of the

Name	Type	Mandatory	Description
			Cardholder billing address associated with the card used for this purchase.
billAddrState	AN3	No	Cardholder Billing Address State. The state or province of the Cardholder billing address associated with the card used for this purchase.
mobilePhone		Yes (if available)	Mobile phone number provided by the Cardholder.
cc	N..3	Yes	Country Code.
subscriber	N..15	Yes	Subscriber
workPhone		Yes (if available)	Work phone number provided by the Cardholder.
cc	N..3	Yes	Country Code.
subscriber	N..15	Yes	Subscriber.
purchaseInstalData	N3	Yes (for instalment payments)	Instalment Payment Data. Indicates the maximum number of authorizations permitted for instalment payments.
merchantRiskIndicator		No	Merchant Risk Indicator. Merchant's assessment of the level of fraud risk for the specific authentication for both

Name	Type	Mandatory	Description
			the cardholder and the authentication being conducted.
deliveryEmailAddress	ANS..254	No	Delivery Email Address For Electronic delivery, the email address to which the merchandise was delivered.
deliveryTimeframe	N2	No	Merchandise delivery timeframe.
giftCardAmount	N..15	No	Gift Card Amount. For a prepaid or gift card purchase, the purchase amount total, in major units.
giftCardCount	N..2	No	Gift Card Count. For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased.
giftCardCurr	N3	No	Gift Card Currency. For prepaid or gift card purchase, the currency code of the card as defined in ISO 4217.
preOrderDate	N8	No	Pre-Order Date. For a pre-ordered purchase, the expected date that the merchandise will be available. The date format is *yyyyMMdd*.
preOrderPurchaseInd	N2	No	Pre-Order Purchase Indicator. Indicates whether the Cardholder is placing an order for merchandise with a future availability or release date.
reorderItemsInd	N2	No	Indicates whether the Cardholder is reordering previously purchased merchandise.

Name	Type	Mandatory	Description
shipIndicator	N2	No	Shipping Indicator that indicates shipping method chosen for the transaction.
messageExtension	ANS..81920	No	Message Extension that passes data necessary to support requirements not defined in the 3-D Secure message.
criticalityIndicator	B	Yes	A Boolean value indicating whether the recipient must be able to interpret the contents of the extension to interpret the entire message.
data	ANS..8059	Yes	The data carried in the extension.
id	ANS..64	Yes	A unique identifier for the extension.
name	ANS..64	Yes	The name of the extension data set as defined by the extension owner.

Response

An HTTP status (with a code) is returned as a response. In the case of a success status (`200`), it is returned with a JSON structure containing ARes.

Name	Type	Mandatory	Description
HTTP/*<version>* *<status_code>*	N3	No	Response code that reflects the status: • 200 – OK This code is returned with a JSON structure. See the description below and an example in Appendix 1: Request and response examples. • 400 – Bad request

The response structure contents

The following parameters are transferred in a JSON structure:

Name	Type	Mandatory	Description
status	ANS..128	Yes	Status of request processing. Possible value: • SUCCESS — ARes successfully received; • INTERNAL_VALIDATION_FAILED — Internal validation of AReq message failed; • INCORRECT_DS_RESPONSE — DS returns Error message or invalid Ares; • NO_CONNECTION_TO_DS — No connection to DS; • INTERNAL_ERROR — Internal error.
statusMessage	ANS..256	No	Status message, for all statuses apart SUCCESS.
packedCReq	ANS..1024	Yes	Base64 packed CReq to send to ACS URL. It will be filled if **TransactionStatus**= C or **acsChallengeMandated** = true.
threeDSServerTransID	AN36	Yes	Unique 3DS Server transaction identifier. It must be passed in a subsequent authentication request.
acsChallengeMandated	B	Yes	ACS Challenge Mandated Indicator. It indicates whether a challenge is required for the transaction to be authorized due to local/regional mandates or other variable. Required if **TransactionStatus** = C.
acsOperatorID	ANS..32	No	ACS identifier by DS assigned.
acsReferenceNumber	ANS..32	Yes	ACS Reference Number. Unique identifier assigned by the EMVCo Secretariat upon Testing and Approval.

Name	Type	Mandatory	Description
acsRenderingType		Yes	ACS Rendering Type. Identifies the ACS UI Template that the ACS will first present to the consumer. It is required if **`TransactionStatus`** `= C`.
acsInterface	N2	Yes	ACS interfaces that is presented to the cardholder in the challenge flow. Values accepted: - `01 = Native UI` - `02 = HTML UI`
acsUiTemplate	N2	Yes	UI Template format that the ACS first presents to the consumer. Valid values for each Interface: - `Native UI = 01–04` - `HTML UI = 01–05.` Values accepted: - `01 = Text` - `02 = Single Select` - `03 = Multi Select` - `04 = OOB` - `05 = HTML Other`
acsSignedContent	ANS..1024	Yes	ACS Signed Content. Contains the JWS object created by the ACS for the ARes message. It is required if **`TransactionStatus`**`=C`.
acsTransID	ANS36	Yes	ACS Transaction ID. Universally Unique transaction identifier assigned by the ACS to identify a single transaction.
acsURL	ANS..2048	Yes	Fully qualified URL of the ACS to be used for the challenge. It is required if **`TransactionStatus`**`=C`.
authenticationType	N2	Yes	Type of authentication method the Issuer will use to challenge the Cardholder. It is

Name	Type	Mandatory	Description
			either used in the ARes message or used by the ACS in the RReq message. It is required if **`TransactionStatus`**=C. Values accepted: • `01 = Static` • `02 = Dynamic` • `03 = OOB`
authenticationValue	ANS28	Yes	Authentication Value. Payment System specific value provided as part of the ACS registration for each supported DS. It is required if **`TransactionStatus`**=Y. A 20-byte value that has been Base64 encoded, giving a 28-byte result.
cardholderInfo	ANS..128	No	Cardholder Information Text. Text provided by the ACS/Issuer to the Cardholder during a Frictionless transaction that was not authenticated by the ACS. The Issuer can optionally provide information to the Cardholder.
dsReferenceNumber	ANS..32	Yes	DS Reference Number. EMVCo-assigned unique identifier to track approved DS.
dsTransID	ANS36	Yes	Universally unique transaction identifier assigned by the DS to identify a single transaction.
eci	N2	Yes	Electronic Commerce Indicator (ECI). Payment System-specific value provided by the ACS to indicate the results of the attempt to authenticate the Cardholder.
messageVersion	ANS..8	Yes	Protocol version identifier. The Message Version Number is set by the 3DS Server.

Name	Type	Mandatory	Description
messageType	A4	Yes	Message Type. Value accepted: • ARes
messageExtension	ANS..81920	No	Message Extension that passes data necessary to support requirements not defined in the 3-D Secure message.
criticalityIndicator	B	Yes	A Boolean value indicating whether the recipient must be able to interpret the contents of the extension to interpret the entire message.
data	ANS..8059	Yes	The data carried in the extension.
id	ANS..64	Yes	A unique identifier for the extension.
name	ANS..64	Yes	The name of the extension data set as defined by the extension owner.
transStatus	A1	Yes	Transaction Status that indicates whether a transaction is qualifies as an authenticated transaction or account verification. Values accepted: • 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; • A = Attempts Processing Performed, Not Authenticated/Verified, but a proof of attempted • authentication/verification is provided;

Name	Type	Mandatory	Description
			• `C = Challenge Required, Additional authentication is required;` • `R = Authentication/Account Verification Rejected, Issuer is rejecting authentication/verification and request that authorization not be attempted.`
transStatusReason	N2	No	Transaction Status Reason. Provides the details on the Transaction Status. It is required if **`TransactionStatus`**`=N`,`U` or `R`. Values accepted: • `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`
sdkTransID	ANS36	Yes for APP	SDK Transaction ID. Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction.

5.3 Service to get a 3DS Method completion notification

To receive notifications about 3DS Method completion from ACS, the **`/api/v1/acs/notification`** method is used.

> **Note:** There are equivalent methods with different names (such as **`/acs/notification`** or **`handleAcsNotification`**). Those methods are obsolete, use the method mentioned above.

The POST method is used to send the request.

Request parameter

Name	Type	Mandatory	Description
threeDSServerTransID	string	Yes	Universally unique transaction identifier assigned by the 3DS Server to a single transaction to identify it.

Response

An HTTP status (with a code) is returned as a response.

Name	Type	Mandatory	Description
HTTP/*<version>* *<status_code>*	N3	No	Response code that reflects the status: • 200 – OK • 400 – Bad request (the required **`threeDSServerTransID`** was not found) • 405 – Wrong method is used to send the request

5.4 Request for RReq by transaction ID

The **`/api/v1/rreq`** method sent with GET is used to receive a model of RReq (the Results Request message) by **`threeDSServerTransID`**.

> **Note:** There are equivalent methods with different names. Those methods are obsolete, use the method mentioned above.

The GET method is used to send the request.

Request parameter

Name	Type	Mandatory	Description
threeDSServerTransID	AN36	Yes	Universally unique transaction identifier assigned by the 3DS Server to a single transaction to identify it.

Response

An HTTP status (with a code) is returned as a response. In the case of a success status (`200`), it is returned with a structure of parameters containing RReq.

Name	Type	Mandatory	Description
HTTP/*<version>* *<status_code>*	N3	No	Response code that reflects the status: • 200 – OK This code is returned with the RReq message model. For the description of RReq parameters see Request body in the section below and an example in Appendix 1: Request and response examples. • 400 – Bad request

5.5 RReq request for RRes

The value for the method is taken by the 3DS Server from the **`config. acquirers.callbackResultUrl`** parameter value in the `application-prod.yml` file (see *3DS Server Installation and Configuration Guide*). When the 3DS Server receives the Result Request from ACS, it forwards it to the URL specified in the parameter (figure 1, step 12).

The **`/api/v1/result`** method with POST is used to process the result request (RReq) and to return the Results Response (RRes):

> **Note:** There are equivalent methods with different names. Those methods are obsolete, use the method mentioned above.

The POST method is used to send the request.

Request body

The request is sent as a body with the RReq (Results Request) contents.

The request body contents

The content type is `application/json`. The charset is UTF-8.

Name	Type	Mandatory	Description
threeDSServerTransID	AN36	Yes	Unique 3DS Server transaction identifier that must be passed in a subsequent authentication request. The maximum length is 36 characters.
messageVersion	ANS..8	Yes	Protocol version identifier. The Message Version Number is set by the 3DS Server. The value length is 5-8 characters.
messageType	A4	Yes	Message Type. Value accepted: ● `RReq`
messageCategory	N2	Yes	Message Category. Identifies the category of the message for a specific use case. Values accepted: ● `01 – Payment Authentication` ● `02 - Non-Payment Authentication`

transStatus	A1	Yes	Transaction Status. Indicates whether a transaction qualifies as an authenticated transaction or account verification. Values accepted: • 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; • A = Attempts Processing Performed, Not Authenticated/Verified, but a proof of attempted authentication/verification is provided; • C = Challenge Required, Additional authentication is required; • R = Authentication/Account Verification Rejected, Issuer is rejecting authentication/verification and request that authorization not be attempted.
acsTransID	ANS36	Yes	ACS Transaction ID. Universally Unique transaction identifier assigned by the ACS to identify a single transaction.
acsRenderingType		Yes	ACS Rendering Type. Identifies the ACS UI Template that the ACS will first present to the consumer. Required if **`TransactionStatus`**=C.

acsInterface	N2	Yes	ACS Interface that is presented to the cardholder in the challenge flow. Values accepted: • 01 = Native UI • 02 = HTML UI
acsUiTemplate	N2	Yes	ACS UI Template. Identifies the UI Template format that the ACS first presents to the consumer. Valid values for each Interface are: • Native UI = 01–04 • HTML UI = 01–05 Values accepted: • 01 = Text • 02 = Single Select • 03 = Multi Select • 04 = OOB • 05 = HTML Other
authenticationType	N2	Yes	Type of authentication method the Issuer will use to challenge the Cardholder. It is either used in the ARes message or used by the ACS in the RReq message. It is required if **TransactionStatus**=Y or N. Values accepted: • 01 = Static • 02 = Dynamic • 03 = OOB
authenticationValue	ANS28	Yes	Authentication Value. Payment System specific value provided as part of the ACS registration for each supported DS. It is required if **TransactionStatus**=Y or A. A 20-byte value that has been Base64 encoded, giving a 28-byte result.

dsTransID	ANS36	Yes	DS Transaction ID. Universally unique transaction identifier assigned by the DS to identify a single transaction.
eci	N2	Yes	Electronic Commerce Indicator (ECI). Payment System specific value provided by the ACS to indicate the results of the attempt to authenticate the Cardholder.
interactionCounter	N2	Yes	Interaction Counter. Indicates the number of authentication cycles attempted by the Cardholder.
challengeCancel	N2	No	Challenge Cancelation Indicator. Indicator informing the ACS and the DS that the authentication has been cancelled. Values accepted: ● `01 = Cardholder selected “Cancel”;` ● `04 = Transaction Timed Out at ACS - other Timeouts;` ● `05 = Transaction Timed Out at ACS—First CReq not received by ACS;` ● `06 = Transaction Error;` ● `07 = Unknown;` ● `08 = Transaction Timed Out at SDK.`

transStatusReason	N2	No	Transaction Status Reason. Provides the details on the Transaction Status. It is required if **`TransactionStatus`**`=N`,`U` or `R`. Values accepted: • `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`
messageExtension	ANS..81920	No	Message Extension that passes data necessary to support requirements not defined in the 3-D Secure message.
criticalityIndicator	B	Yes	A Boolean value indicating whether the recipient must be able to interpret the contents of the extension to interpret the entire message.
data	ANS..8059	Yes	The data carried in the extension.
id	ANS..64	Yes	A unique identifier for the extension.

name	ANS..64	Yes	The name of the extension data set as defined by the extension owner.

Response

An HTTP status (with a code) is returned as a response. In the case of a success status (`200`), it is returned with a structure of parameters detailing the RRes message. In case of an error, the error details are returned.

Name	Type	Mandatory	Description
HTTP/*<version>* *<status_code>*	N3	No	Response code that reflects the status: ● 200 – OK This code is returned with a JSON structure. See the description below. ● 400 – Bad request

The response structure contents

The following RRes parameters are transferred in a JSON structure:

Name	Type	Mandatory	Description
threeDSServerTransID	AN36	Yes	Unique 3DS Server transaction identifier to identify a single transaction. The maximum length is 36 characters. If 3DS Method was previously invoked, the sent **`threeDSServerTransID`** must be used. If 3DS Method was not invoked, 3DS Server will generate a new transaction identifier.
acsTransID	ANS36	Yes	Universally unique transaction identifier assigned by the ACS to identify a single transaction.

Name	Type	Mandatory	Description
			The maximum length is 36 characters.
dsTransID	ANS36	Yes	Universally unique transaction identifier assigned by the DS to identify a single transaction. The identifier format is IETF RFC 4122. The maximum length is 36 characters.
messageType	A4	Yes	Message Type. Value accepted: ● `RRes`
messageVersion	ANS..8	Yes	Protocol version identifier. The Message Version Number is set by the 3DS Server.
resultsStatus or **ResultsStatus**	N2	Yes	Status of the Results Request message from the 3DS Server to provide additional data to ACS. This status indicates if the message was successfully received for further processing or provides the details on why the Challenge could not be completed from the 3DS Client to the ACS. The possible values: ● `01` — RECEIVED, Results Request received for further processing ● `02` — CHALLENGE_REQUEST_NOT_SENT_TO_ACS, Challenge Request not sent to ACS by 3DS Requestor (3DS Server or 3DS Requestor opted out of the challenge) ● `03` — ERROR_ARES_DELIVERY, ARes challenge data not delivered to the 3DS Requestor due to a technical error. ● 04–79 — Reserved for EMVCo future use (the values are invalid until defined by EMVCo).

Name	Type	Mandatory	Description
			• 80-99 — Reserved for the DS use.

5.6 CRes request

After receiving the corresponding RRes message, ACS generates the CRes (Challenge Response) message and invokes the browser to send an HTTP POST to a redirect destination. This action completes the Challenge procedure (when additional interaction with the cardholder is required to complete the authentication).

The **`sendChallengeToGateway`** parameter in `application-prod.yml` defines the redirect destination.

- `false` — redirect 303 is used (which uses only the GET method for redirecting). For more information, see “303 See Other” in *Documentation for Web developers* at https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/303. In this case the browser is redirected to the **`notificationURL`**.
- `true` — redirect 307 is used (which uses the original method for redirecting without changing it: GET, POST and so on). For more information, see “307 Temporary Redirect” in *Documentation for Web developers* at https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/307. In this case the cardholder’s browser is redirected to the `finish.html` page. This value is used if 3DS Server is integrated with the payment gateway (EPG).

The default value for **`sendChallengeToGateway`** is `false`.

The **`/api/v1/challenge`** method described below is used to require the Challenge Response.

Note: There are equivalent methods with different names. Those methods are obsolete, use the method mentioned above.

The POST method is used to send the request.

Request parameter

Name	Type	Mandatory	Description
cres	string	Yes	CRes message that is the ACS response to the CReq message (Base64 encoded). It can indicate the result of the Cardholder authentication or, in the case of an App-based model, signal that further Cardholder interaction is required to complete the authentication.

The cres structure contents

The **`cres`** parameter transfers a Base64 encoded string, which when it is decoded, is mapped into a structure that contains the following fields:

Name	Type	Mandatory	Description
threeDSServerTransID	AN36	Yes	Unique 3DS Server transaction identifier that must be passed in a subsequent authentication request.
acsTransID	ANS36	Yes	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.
messageType	A4	Yes	Message Type. Value accepted: • `CRes`
messageVersion	ANS..8	Yes	Protocol version identifier. The Message Version Number is set by the 3DS Server.
transStatus	A1	Yes	Transaction Status that indicates whether a transaction qualifies as an authenticated transaction or account verification. It is present only in the final CRes message. Values accepted: • `Y = Authentication/Account Verification Successful;` • `N = Not Authenticated/Account Not Verified, Transaction denied.`
sdkTransID	ANS36	Yes for APP	Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction. The maximum length is 36 characters.
acsCounterAtoS	N3	Yes	ACS to SDK Counter used as a security measure in the secure channel for communication from ACS to 3DS SDK. The maximum length is 3 characters.

Name	Type	Mandatory	Description
acsHTML	5 (HTML)	Conditional upon the selection of the ACS UI	HTML provided by ACS in the CRes message. It is used when HTML is specified in the ACS UI Type during the cardholder challenge. The maximum size of the file is 100KB This value will be Base64url encoded before being included in the CRes message.
acsHTMLRefresh	5 (HTML)	No	HTML provided by ACS in the CRes message to be used in the Out-of-Band flow when the HTML is specified in the ACS UI Type during the cardholder challenge. If **`acsHTMLRefresh`** is present in the CRes message, the SDK will display it when the application is moved to the foreground. The maximum size of HTML is 100KB This value will be Base64url encoded prior to being placed into the CRes message.
AcsUiType or **acsUiType**		Yes	User interface type that the 3DS SDK will render, which includes the specific data mapping and requirements.
challengeAddInfo	ANS 1..256	No	Information Text provided by ACS or Issuer to the cardholder during an OOB authentication to replace **`challengeInfoText`** and **`challengeInfoTextIndicator`** (described below) in the OOB Template. If the field is populated, this information is displayed to the cardholder by the SDK when the 3DS Requestor App is operating.

Name	Type	Mandatory	Description
challengeCompletionInd	Boolean	Yes	Indicator for the state of the ACS challenge cycle, signifying whether the challenge has completed or will require additional messages. It must be added to all CRes messages to convey the current state of the transaction. The available values are: • Y — ACS will populate the Transaction Status in the CRes message N — the parameter is not added to CRes
challengeInfoHeader	ANS 1..45	No	Header text for the challenge information screen that is can be displayed to the cardholder.
challengeInfoLabel	ANS 1..45	No	Label used to modify the Challenge Data Entry field provided by the Issuer.
challengeInfoText	ANS 1..350	No	Text provided by the ACS or Issuer to the cardholder during the Challenge message exchange.
challengeInfoTextIndicator	Boolean	No	Text Indicator that indicates when the Issuer or ACS needs to add a warning icon or similar visual indicator to draw the cardholder's attention to the text from **`challengeInfoText`** displayed.
challengeSelectInfo	ANS 1..45	No	Expandable Information Label displayed to the cardholder for expanding the contents in **`expandInfoText`**.

Name	Type	Mandatory	Description
expandInfoText	ANS 1..256	No	Expandable information text provided by the Issuer from ACS to be displayed to the cardholder for additional information and the format will be an expandable text field
issuerImage		Conditional (depends on IPN)	Issuer image sent in the initial CRes message from ACS to the 3DS SDK to provide the URL(s) of the Issuer logo to be used in the Native UI. The presence of this field is a payment system specific
messageExtension	ArrayList 81920	Conditional (depends on DS)	Message extension contains the data necessary to support the requirements otherwise defined in the 3-D Secure message. The conditions of usage to be set by each DS.
oobAppURL	ANS 1..256	No	Mobile deep link to an authentication app used in the out-of-band authentication. The App URL will open the appropriate location within the authentication app.
oobAppLabel	ANS 1..45	No	Label to be displayed for the link to the OOB App URL. Example: `“oobAppLabel”: “Click here to open Your Bank App”`
oobContinueLabel	ANS 1..45	Conditional (Yes if when ACS UI Type = 04)	Label to be used in the UI for the button that the cardholder selects when they have completed the OOB authentication.

Name	Type	Mandatory	Description
psImage		Conditional (depends on IPN)	Payment system image sent in the initial CRes message from ACS to the 3DS SDK to provide the URL(s) of the DS or payment system logo to be used in the Native UI.

Response parameter

An HTTP status (with a code) is returned as a response.

Name	Type	Mandatory	Description
HTTP/*<version>* ***<status_code>***	N3	No	Response code that reflects the status: • 303 – redirect to the notification URL • 307 – redirect to finish.html • 500 – any error

5.7 Browser data collection request

The `/api/v1/client/gather` method described below returns the HTML-page used for collection of the data about the cardholder's browser and device when a challenge (additional cardholder's interaction) is necessary. The HTML-page is displayed through the iframe.

Note: There are equivalent methods with different names. Those methods are obsolete, use the method mentioned above.

The POST method is used to send the request.

Request parameter

Name	Type	Mandatory	Description
threeDSServerTransID	string	Yes	Universally unique transaction identifier assigned by the 3DS Server to a single transaction to identify it.

Response

An HTTP status (with a code) is returned as a response. In the case of a success status (`200`), it is returned with an HTML-page for the browser information collection.

Name	Type	Mandatory	Description
HTTP/*<version>* *<status_code>*	N3	No	Response code that reflects the status: • 200 – OK This code is returned with an HTML-page for the information collection from the browser in the UTF-8 format. • 500 – any error

5.8 Browser data transfer request

The **`/api/v1/client`** method described below is used to transmit the cardholder's browser information and device details when a challenge (additional cardholder's interaction) is necessary.

Note: There are equivalent methods with different names. Those methods are obsolete, use the method mentioned above.

The POST method is used to send the request.

Request parameter

Name	Type	Mandatory	Description
clientInfo	string	Yes	Browser information and device details. See the **`clientInfo`** parameter description below.
threeDSServerTransID	string	Yes	Universally unique transaction identifier assigned by the 3DS Server to a single transaction to identify it.

The clientInfo structure contents

The **`clientInfo`** parameter contains the following JSON structure:

Name	Type	Manda-tory	Description
userAgent	String	Yes	Information the cardholder's web browser sends in the **`User-Agent`** HTTP header when making requests to web sites. It is a string containing information about the cardholder's browser, operating system, device type and other details. The **`User-Agent`** format varies for different browsers.
colorDepth	String	Yes	Bit depth of the color palette for displaying images on the cardholder's device screen.
screenHeight	Integer	Yes	Height of the cardholder's device screen.
screenWidth	Integer	Yes	Width of the cardholder's device screen.
javaEnabled	boolean	Yes	Parameter that specifies whether supporting Java is enabled for the cardholder's browser. The available values are: • `true` `false`
browserLanguage	String	Yes	Language of the cardholder's browser.
browserTimeZoneOffset	Integer	Yes	Difference between UTC time and the cardholder's browser local time, in minutes.
browserAcceptHeader	String	Yes	Parameter that informs the server, to which the browser sends a request, on what file formats (MIME-types) are acceptable for the browser as a response.
browserIpAddress	String	Yes	IP address of the cardholder's browser.
fingerprint	String	No	Information collected from the device browser for subsequent identification.

Name	Type	Mandatory	Description
OS	String	No	Operating system used by the cardholder's device.
OSVersion	String	No	Version of the operating system used by the cardholder's device.
device	String	No	Information about the cardholder's device (model, version, and so on).
deviceType	String	No	Type of device on which the browser is running (mobile phone, desktop, tablet, and so on).
isMobile	boolean	No	Parameter that specifies whether the cardholder's device is mobile. The available values are: • `true` • `false`
screenPrint	String	No	Information on the the cardholder's device screen resolution.
plugins	String	No	List of plug-ins installed to the cardholder's device browser.
browserTimeZone	String	No	Time zone of the cardholder's browser.

See also Browser data example.

Response

An HTTP status (with a code) is returned as a response.

Name	Type	Mandatory	Description
HTTP/*<version> <status_code>*	N3	No	Response code that reflects the status: • 200 – OK

Name	Type	Mandatory	Description
			• 500 – any error

5.9 Error data collection request

The **`/api/v1/client/errors`** method described below is used to receive notifications about errors on the cardholder's browser side.

> **Note:** There are equivalent methods with different names. Those methods are obsolete, use the method mentioned above.

The POST method is used to send the request.

Request parameters

Name	Type	Mandatory	Description

Body with the JSON structure		Yes	Body with the following contents (a JSON parameters structure):
			`{`
			`"message": "string",`
			`"name": "string",`
			`"stack": "string"`
			`}`

threeDSServerTransID	string	Yes	Universally unique transaction identifier assigned by the 3DS Server to a single transaction to identify it.

Response

An HTTP status (with a code) is returned as a response.

Name	Type	Mandatory	Description
HTTP/*<version>* *<status_code>*	N3	No	Response code that reflects the status: • 200 – OK

Name	Type	Mandatory	Description
			• 400 – Bad request

Appendix 1: Request and response examples

This section contains examples of some types of requests.

Authentication Request body example

```
{
  "acctID": "string",
  "acctInfo": {
    "chAccAgeInd": "01",
    "chAccChange": "yyyyMMdd",
    "chAccChangeInd": "01",
    "chAccDate": "yyyyMMdd",
    "chAccPwChange": "yyyyMMdd",
    "chAccPwChangeInd": "01",
    "nbPurchaseAccount": "string",
    "paymentAccAge": "yyyyMMdd",
    "paymentAccInd": "01",
    "provisionAttemptsDay": "string",
    "shipAddressUsage": "yyyyMMdd",
    "shipAddressUsageInd": "01",
    "shipNameIndicator": "01",
    "suspiciousAccActivity": "01",
    "txnActivityDay": "string",
    "txnActivityYear": "string"
  },
  "acctNumber": "string",
  "acctType": "01",
  "acquirerBIN": "string",
  "acquirerMerchantID": "string",
  "addrMatch": true,
  "billAddrCity": "string",
  "billAddrCountry": "string",
  "billAddrLine1": "string",
"billAddrLine2": "string",
"billAddrLine3": "string",
"billAddrPostCode": "string",
"billAddrState": "string",
"broadInfo": {},
"browserAcceptHeader": "string",
"browserColorDepth": "1",
"browserIP": "string",
"browserJavaEnabled": true,
"browserLanguage": "string",
"browserScreenHeight": "string",
"browserScreenWidth": "string",
"browserTZ": "string",
"browserUserAgent": "string",
"cardExpiryDate": "yyMM",
"cardholderName": "string",
"deviceChannel": "01",
"deviceInfo": "string",
"deviceRenderOptions": {
 "sdkInterface": "01",
 "sdkUiType": [
  "01"
 ]
},
"dsEmulator": true,
"dsReferenceNumber": "string",
"dsTransID": "string",
"dsURL": "string",
"email": "string",
"homePhone": {
 "cc": "string",
 "subscriber": "string"
},
"mcc": "string",
"merchantCountryCode": "string",
"merchantName": "string",
"merchantRiskIndicator": {
 "deliveryEmailAddress": "string",
 "deliveryTimeframe": "01",
 "giftCardAmount": "string",
 "giftCardCount": "string",
 "giftCardCurr": "string",
 "preOrderDate": "yyyyMMdd",
 "preOrderPurchaseInd": "01",
 "reorderItemsInd": "01",
 "shipIndicator": "01"
},
"messageCategory": "01",
"messageExtension": [
 {
  "criticalityIndicator": true,
  "data": {},
  "id": "string",
  "name": "string"
 }
],
"messageType": "AReq",
"messageVersion": "string",
"mobilePhone": {
 "cc": "string",
 "subscriber": "string"
},
"notificationURL": "string",
"payTokenInd": true,
"purchaseAmount": "string",
"purchaseCurrency": "string",
"purchaseDate": "yyyyMMddHHmmss",
"purchaseExponent": "string",
"purchaseInstalData": "string",
"recurringExpiry": "yyyyMMdd",
"recurringFrequency": "string",
"returnUrl": "string",
"sdkAppID": "string",
"sdkEncData": "string",
"sdkEphemPubKey": {
 "alg": "string",
 "crv": "string",
 "d": "string",
 "key_ops": "string",
 "kid": "string",
 "kty": "string",
 "use": "string",
 "x": "string",
 "x5c": "string",
 "x5t": "string",
 "x5u": "string",
 "y": "string"
},
"sdkMaxTimeout": "string",
"sdkReferenceNumber": "string",
"sdkTransID": "string",
"shipAddrCity": "string",
"shipAddrCountry": "string",
"shipAddrLine1": "string",
"shipAddrLine2": "string",
"shipAddrLine3": "string",
"shipAddrPostCode": "string",
"shipAddrState": "string",
"threeDSCompInd": "Y",
"threeDSRequestorAuthenticationInd": "01",
"threeDSRequestorAuthenticationInfo": {
 "threeDSReqAuthData": "string",
 "threeDSReqAuthMethod": "01",
 "threeDSReqAuthTimestamp": "yyyyMMddHHmm"
},
"threeDSRequestorChallengeInd": "01",
"threeDSRequestorID": "string",
"threeDSRequestorName": "string",
"threeDSRequestorPriorAuthenticationInfo": {
 "threeDSReqPriorAuthData": "string",
 "threeDSReqPriorAuthMethod": "01",
 "threeDSReqPriorAuthTimestamp": "yyyyMMddHHmm",
 "threeDSReqPriorRef": "string"
},
 "threeDSRequestorURL": "string",
 "threeDSServerOperatorID": "string",
 "threeDSServerRefNumber": "string",
 "threeDSServerTransID": "string",
 "threeDSServerURL": "string",
 "threeRIInd": "01",
 "transType": "01",
 "workPhone": {
   "cc": "string",
   "subscriber": "string"
 }
}
```

Authentication Response body example

```
{
  "acsChallengeMandated": true,
  "acsOperatorID": "string",
  "acsReferenceNumber": "string",
  "acsRenderingType": {
    "acsInterface": "01",
    "acsUiTemplate": "01"
  },
  "acsSignedContent": "string",
  "acsTransID": "string",
  "acsURL": "string",
  "authenticationType": "01",
  "authenticationValue": "string",
  "broadInfo": {},
  "cardholderInfo": "string",
  "dsReferenceNumber": "string",
  "dsTransID": "string",
  "eci": "string",
  "messageExtension": [
    {
      "criticalityIndicator": true,
      "data": {},
      "id": "string",
      "name": "string"
    }
  ],
  "messageType": "AReq",
  "messageVersion": "string",
  "packedCReq": "string",
  "sdkTransID": "string",
  "status": "SUCCESS",
  "statusMessage": "string",
  "threeDSServerTransID": "string",
  "transStatus": "Y",
  "transStatusReason": "01"
}
```

Results Request body example

```
{
  "acsRenderingType": {
    "acsInterface": "01",
    "acsUiTemplate": "01"
  },
  "acsTransID": "string",
  "authenticationMethod": "01",
  "authenticationType": "01",
  "authenticationValue": "string",
  "challengeCancel": "01",
  "dsTransID": "string",
  "eci": "string",
  "interactionCounter": "string",
  "messageCategory": "01",
  "messageExtension": [
    {
      "criticalityIndicator": true,
      "data": {},
      "id": "string",
      "name": "string"
    }
  ],
  "messageType": "AReq",
  "messageVersion": "string",
  "threeDSServerTransID": "string",
  "transStatus": "Y",
  "transStatusReason": "01"
}
```

Browser data example

```
{
"userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/69.0.3497.100 Safari/537.36",
"fingerprint": 2577441658,
"OS": "Windows",
"OSVersion": "10",
"isMobile": false,
"screenPrint": "Current Resolution: 1536x864, Available Resolution: 1536x834, Color Depth: 24, Device XDPI:
undefined, Device YDPI: undefined",
"plugins": "Chrome PDF Plugin, Chrome PDF Viewer, Native Client",
"javaEnabled": false,
"browserLanguage": "en-US",
"browserTimeZone": "Europe/Moscow"
}
```

Glossary

Term	Description
3DS	3-D Secure
3DS Integrator	An EMV 3-D Secure participant that facilitates and integrates the 3DS Requestor Environment, and optionally facilitates integration between the Merchant and the Acquirer.
3DS Requestor	The initiator of the EMV 3-D Secure Authentication Request. For example, this may be a merchant or a digital wallet requesting authentication within a purchase flow.
3DS Requestor Environment	The 3DS Requestor-controlled components (3DS Requestor App, 3DS SDK, and 3DS Server) are typically facilitated by the 3DS Integrator. Implementation of the 3DS Requestor Environment will vary as defined by the 3DS Integrator.
3DS Client	Component on a Consumer Device that initiates a 3-D Secure authentication.
3DS Server	Server that is necessary to enable 3-D Secure 2 checks when processing payments from merchants' sites. This server handles online transactions and facilitates communication between the 3DS Requestor (a merchant or e-wallet) and the DS.
ACS	Access Control Server
DS	Directory Server
EPG	E-commerce Payment Gateway

References

- *EMV 3-D Secure Protocol and Core Functions Specification v 2.1.0*