Upgrade to 3DSV2

In October 2022, all transactions will only be auhtenticated with 3DSV2. 3DSV1 authentication won't be available anymore.

This section is a synthesis gathering information for merchants upgrading from 3DSV1 to 3DS2

For Axepta, it means : Add the parameter 'msgVer=2.0' for one-shot payments. Other payment use-cases are described in the section Payment Features

Are you concerned ?

To verify the version of the 3-D Secure protocol used by your online store, please refer to Identify your 3D Secure version.

Introduction

3D Secure authentication protects the merchant against "cardeholder challenge".

3D Secure Version 2 provides a mechanism for strong authentication in accordance with PSD2.

3DS V2 allows frictionless payments to cardholders, in this case the issuing bank (bank of the cardholder) can approve a transaction without explicit action done by the cardholder (passive authentication).

By default, payment requests are sent with “nopreference”, the merchant does not specify a preference for the authentication mode (SCA or passive authentication).

More details :

- Global overview : 3DSV2 and frictionless
- Implementation : Exemption & 'Frictionless' payments
- Liability shift : Liability shift and 3DS Matrix

Scope

CB (Carte Bancaire), Visa, Mastercard and AMEX transactions.

Upgrade to 3DSV2 - With CMS

Find all the information in the section: Get your 3DSecure 2.0 plugin for free

Upgrade to 3DSV2 - Direct Integration

Synthesis

3DSV2 upgrade requires to manage :

- **New parameters in the request and responses**
  - **Parameter 'msgVer=2.0'**
  - Optionnal Fields (NVP values) and JSON objects – for dedicated use-cases
  - Additionnal authentication data
- **GET & POST responses** for URLSuccess / URL Failure / URL Notify
- **New 3DS Response Codes**

For one-shot payments, upgrading to 3DSV2 means :

- Add parameter "msgVer=2.0" in the payment request
- Manage POST responses

Detailed overview

Use-case	3DSV1	3DSV2	Examples : JSON objects to add in ECOM payments
One-Shot payment	-	Add "MsgVer=2.0" - Create a payment request (CIT) *Additional data can be added to the request in order to increase frictionless payments.* *More details :* • *3DSV2 and frictionless* • *Exemption & 'Frictionless' payments*	
Response format **Card payment**	GET for URLFailure / URLSuccess / URLNotify	POST for URLFailure / URLSuccess / URLNotify GET for URLFailure / URLSuccess / URLNotify if 3DSV1 fallback	
Response format **Alternative payment methods**	GET for URLFailure / URLSuccess / URLNotify	GET for URLFailure / URLSuccess / URLNotify	
Card data	PCNr CCExpiry CCCVC CCBrand	JSON Object Card Or JSON Object Card + PCNR / CCExpiry / CCBrand if 3DSV1 fallback	
Use-cases			

One-click
 	
Initial payment
	/	Use "MsgVer=2.0", JSON objects credentialOnFile and threeDSPolicy
		
		One-click payment
			JSON Object credentialOnFile
			
			`{ "type": { "unscheduled": "CIT" }, "initialPayment": true, "useCase": "cof" }`
			
			JSON Object threeDSPolicy
			
			`{ "challengePreference": "mandateChallenge" }`

One-click **One-click payments**	PCNr CCExpiry CCCVC CCBrand	Use "MsgVer=2.0", JSON objects credentialOnFile and threeDSPolicy One-click payment	JSON Object credentialOnFile `{ "type": { "unscheduled": "CIT" }, "initialPayment": false, "useCase": "ucof" }`

<table>
<tr>
<td>Subscription with <u>fixed</u> a mount and frequency

Initial payment</td>
<td>Parameter "RTF=I"

Card data : PCNr, CCExpiry, CCBrand…</td>
<td>Server-to-Server

Use "MsgVer=2.0", JSON objects credentialOnFile and threeDSPolicy

JSON Object Card and the parameter schemeReferenceID are sent in the response.

The object Card available in the response should be decrypted and stored.

The object card used in the next request requires less parameters than the card object in the reponse.

• card:request EN
• card:response EN

Batch

Batch implementation remains the same

<u>More details</u>

• Recurring card payments (Subscription)
• Recurring card payments (Subscription) - MOTO (Mail Order / Telephone Order)
• Create Batch payments (File)</td>
<td>JSON object credentialOnFile

<pre>{
    "type": {
        "recurring": {

"recurringFrequency": 30,

"recurringStartDate":
"2019-09-14",

"recurringExpiryDate":
"2020-09-14"
        }
    },
    "initialPayment":
true,
    "useCase": "fixed"
}</pre>
JSON object threeDSPolicy

<pre>{

"challengePreference":
"mandateChallenge"
}</pre></td>
</tr>
<tr>
<td>Subscription with <u>fixed</u> a mount and frequency

Subscription payment</td>
<td>Parameter "RTF=R"

Card data : PCNr, CCExpiry, CCBrand…</td>
<td>Server-to-Server

Use "MsgVer=2.0", JSON objects credentialOnFile, Card et the parameter schemeReferenceID.

JSON Object Card and the schemeReferenceID are sent in the response.

The object Card available in the response should be decrypted and stored.

The object card used in the next request requires less parameters than the card object in the reponse.

• card:request EN
• card:response EN

Batch

Batch implementation remains the same

<u>More details</u>

• Recurring card payments (Subscription)
• Recurring card payments (Subscription) - MOTO (Mail Order / Telephone Order)
• Create Batch payments (File)</td>
<td>JSON Object credentialOnFile

<pre>{
    "type": {
        "recurring": {

"recurringFrequency": 30,

"recurringStartDate":
"2019-09-14",

"recurringExpiryDate":
"2020-09-14"
        }
    },
    "initialPayment":
false,
    "useCase": "fixed"
}</pre></td>
</tr>
</table>

<table>
<tr>
<td>Subscription with variable amount and frequency

Initial payment</td>
<td>Parameter "RTF=E"

Card data : PCNr, CCExpiry, CCBrand…</td>
<td>Server-to-Server

Use "MsgVer=2.0”, JSON objects credentialOnFile and threeDSPolicy

JSON Object Card and the parameter schemeReferenceID are sent in the response.

The object Card available in the response should be decrypted and stored.

The object card used in the next request requires less parameters than the card object in the reponse.

• card:request EN
• card:response EN

Batch

Batch implementation remains the same

More details

• Recurring card payments (Subscription)
• Recurring card payments (Subscription) - MOTO (Mail Order / Telephone Order)
• Create Batch payments (File)</td>
<td>JSON Object credentialOnFile

<pre>{
    "type": {
        "unscheduled":
"CIT"
    },
    "initialPayment":
true,
          "useCase":
"ucof"
}</pre>
JSON Object threeDSPolicy

<pre>{

"challengePreference":
"mandateChallenge"
}</pre></td>
</tr>
<tr>
<td>Subscription with variable amount and frequency

Subsequent payments</td>
<td>Parameter "RTF=M"

Card data : PCNr, CCExpiry, CCBrand…</td>
<td>Server-to-Server

Use "MsgVer=2.0”, JSON objects credentialOnFile, Card et the parameter schemeReferenceID.

JSON Object Card and the schemeReferenceID are sent in the response.

The object Card available in the response should be decrypted and stored.

The object card used in the next request requires less parameters than the card object in the reponse.

• card:request EN
• card:response EN

Batch

Batch implementation remains the same

More details

• Recurring card payments (Subscription)
• Recurring card payments (Subscription) - MOTO (Mail Order / Telephone Order)
• Create Batch payments (File)</td>
<td>JSON Object credentialOnFile

<pre>{
    "type": {
        "unscheduled":
"MIT"
    },
    "initialPayment":
false,
          "useCase":
"ucof"
}</pre></td>
</tr>
</table>

Payment features are detailed in the section Payment Features

Test your 3DSV2 integration

To test your 3DSV2 integration, you can use our 3DSV2 Test environment

3DSV2 key parameters

Key parameters are used according to the use-cases (one-shot, one-click, recurring...) :

Parameter	Type	Details	Request	Response
msgVer	Parameter	3DSV2 payment request	X	
threeDSPolicy	JSON Object	Manage the authentication request (mandate, challenge...)	X	
browserInfo	JSON Object	Need for paynow.aspx (one-click)	X	
credentialOnFile	JSON Object	Needed if a merchant wants to reuse a card (one-click, recurring...)	X	
billToCustomer	JSON Object	Customer billed for his purchases	X	
shipToCustomer	JSON Object	Customer who received the goods	X	
billingAddress	JSON Object	Billing address	X	
shippingAddress	JSON Object	Shipping address	X	
Card	JSON Object	Card object with the PCNr, card brand, expiry date...	X	X
schemeReferenceId	Parameter	Chaining data used for subscription transactions / recurring payments	X	X
resultsResponse	JSON Object	authentication additionnal data		X
threeDSData	JSON Object	Authentication data		X