Google Pay™

Integration Guide

for Smartpay Advance payment gateway
Web Payment SOAP API

Version 02

12 April 2022

Table of Contents

1 Google Pay™

Google Pay™ makes it easier for your customers to checkout, allowing them to pay with any payment card stored in their Google Pay digital wallet. You can either add the Google Pay button to your checkout page or you may use the Barclaycard Secure Hosted Payment Page. Google Pay is supported on all major browsers. When the customer clicks on the Google Pay button they are presented with a list of payment cards stored in their Google account. They can then select a card and Google will return an encrypted payment token to be used for payment in Smartpay in place of the card detail fields.

This document is intended for merchants using the Barclaycard Smartpay Advance payment gateway who want to take eCommerce payments from Google Pay™ digital wallets using the Web Payment SOAP API. For full information on the overall payment process for Barclaycard Smartpay Advance, and the other features of the interface, you may download the *Web Payment Web Service Integration Guide* here (PDF)

Google Pay is supported with Web Payments API version 30 and above.

Smartpay Integration methods

- *Secure Hosted Payment Page* - Smartpay manages all communications with Google Pay and automatically decrypts payload before processesing authorisation request.
- *Google Pay direct* - Merchant requests encrypted payload directly from Google Pay, passes it to Smartpay for decryption before it processes authorisation request.

Smartpay supports the following Google Pay features:

Google Pay Integration methods

- Web integration

Payment type

- CARD

Authentication methods

- PAN_ONLY
- CRYPTOGRAM_3DS

Card networks

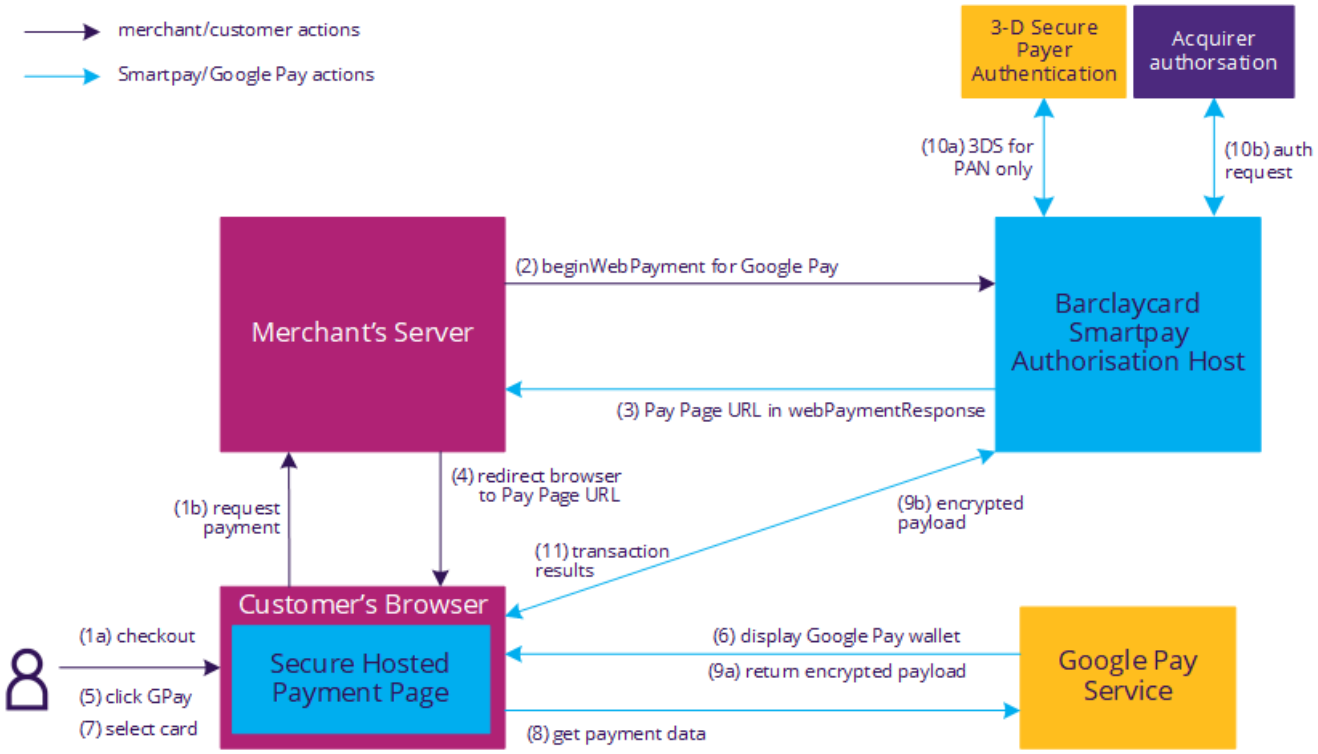
- AMEX
- MASTERCARD
- VISA

The following sections describe how to integrate either using the Hosted Payment Page, or alternativley integrating directly with Google Pay. The former being the simplest approach where you need only send messages to Smartpay; wheras the latter gives you greater control but requires you to set up a Google Pay merchant account and submit messages to both Google Apy API and the Smartpay Web Payments API.

2 Integrating to Google Pay via Hosted Pay Payment Page

Using the Hosted Payment Page for taking Google Pay payments is much simpler than the direct integration method described below. You only have to integrate directly with Smartpay, and you don't need a Google Pay merchant account or integrate directly with Google Pay. Smartpay and the Hosted Payment Page will manage all necessary communications with Google Pay on your behalf.

Please note that the allowed network cards must be configured in your merchant configuration by your Barclaycard account manager. They can also specify the style of Google Pay buton that appears on the Payment Page. If the customer has cards in their wall et that are not in the 'allowed' list then those cards will be greyed-out/disabled in their wallet.



1. Customer selects the payment button on merchant checkout page.
2. Merchant server sends a request message to Smartpay by calling the beginWebPayment method with fields including the *GooglePay* payment method, authorisation type, and your store result page URL.
3. Initial response will include fields such as the transaction status and redirection URL to the Payment Page.
4. Redirect the customer's browser to the Payment Page. This contains an embedded iFrame that will display the various payment options available, but first Smartpay will query the Google Pay API to determine the user's ab ility to pay (e.g the browser supports Google Pay, and the user either has a saved payment method or can add one). If the user is able to pay using Google Pay then the Payment Page will display the Google Pay button; otherwise, it will not be displayed.
5. Customer clicks the Google Pay button.
6. Google Pay displays the customer's digital wallet.
7. Customer selects a card to be used for payment.
8. Payment Page requests Google Pay payment data.
9. Google Pay returns payment data object containing the encrypted payload; Payment Page passes this to Smartpay.
10. Smartpay decrypts the Google Pay payload to extract the payment credential (either PAN or cryptogram), then sends transaction details first to 3DS if it's a PAN then to the acquirer for authorisation; or for a cryptogram, directly to authorisation. Fraud checking may also be performed if it's a PAN (not shown in diagram), but if it's a cryptogram then this will be skipped.
11. Transaction results are returned to Payment Page.

2.1 Smartpay Request

In order to process a Google Pay transaction be sure to include the following fields in the webPaymentRequest, in addition to the standard payment request fields:

- paymentMethod = *GooglePay*
- authType = *AuthOnly* or *AuthAndSettle*
- storeResultPage = your merchant page that displays the transaction results
- version = *30*, or above

2.2 Smartpay Response

Providing you set the version to 30, or above, in the request then the response will contain these additional fields:

```
<configParams>
    <googlePayGatewayMerchantId> MerchantIdentifier </googlePayGatewayMerchantId>
    <googlePayMerchantId> 12345678901234567890 </googlePayMerchantId>
</configParams>
```

3 Integrating Directly with Google Pay

If you're not using the Hosted Payment Page for taking Google Pay payments then you'll need to integrate directly with Google Pay in order to retrieve the card details in the form of an encrypted payload, and submit to Smartpay in the payment request. But first you must set up your integration details with Google Pay and request production access.

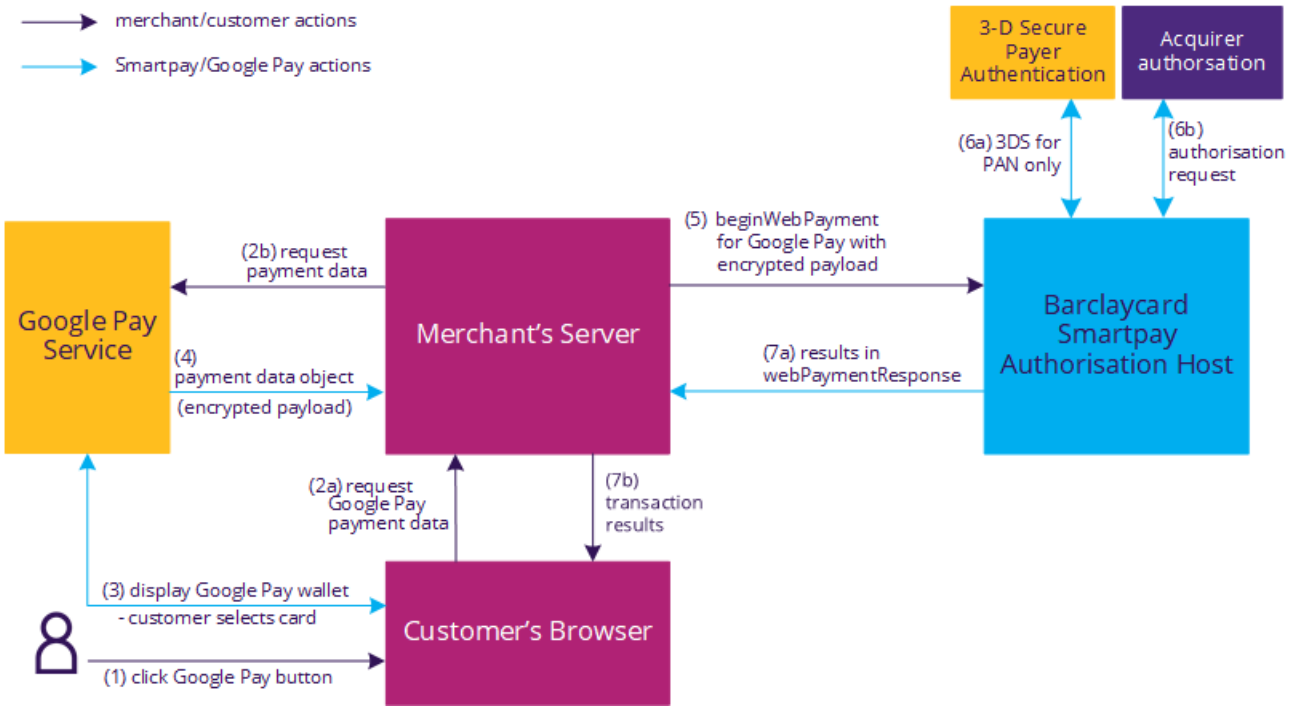
3.1 Getting Started with Google Pay API

Before you can start using the Google Pay API you must perform the following steps:

1. Review and adhere to the Google Pay API Terms of Service and Acceptable Use Policy
2. Review and adhere to the Google Pay Web Brand Guidelines
3. Complete the tutorial and integration checklist Web integration tutorial and Web Integration Checklist
4. Providing you've fulfilled all criteria in the Integration Checklist then you may request production access through the Business Console.
   a. Enter a Business Profile to identify your business with Google. You can enter information such as a business logo, name, support phone numbers or websites.
   b. Add your integration type as 'Gateway web'.
   c. Upload screenshots of your payment flow as proof you've followed the brand guidelines in order to request production access to the Google Pay API.
   d. Once approved Google Pay will assign you a **merchantId**, which will be displayed under your account's *Public merchant profile* setting. You must include this in any requests you send to the Google Pay API.
   e. It's also advisable to add at least one additional user for the Business Console.
5. Once registered, click the *Google Pay API* tab in the Business Console to get started.

3.2 Direct Integration Payment Flow

Please refer to Google Pay's developer documentation, mentioned above, for instructions on how to add a Google Pay payment button to your checkout page; and how to create a payment data request object in order to retrieve the customer's stored card details in an encrypted payload.



1. Customer clicks Google Pay button on merchant checkout page.
2. Merchant server requests Google Pay payment data object from the Google Pay service.
3. Google Pay displays the customer's digital wallet in the browser and the customer selects one of their stored cards to be used for payment.
4. Google Pay returns payment data object containing the encrypted payload, which contains all the card details.
5. Merchant server sends a request message to Smartpay by calling the beginWebPayment method requesting a Google Pay payment with the encrypted payload.
6. Smartpay decrypts the Google Pay payload to extract the payment credentials (either PAN or cryptogram), then sends transaction details first to 3DS if it's a PAN then to the acquirer for authorisation; or for a cryptogram, directly to authorisation. Fraud checking may also be performed if it's a PAN (not shown in diagram), but if it's a cryptogram then this will be skipped.
7. Authorisation response is returned to merchant who informs customer of the transaction results.

3.3 Requesting Google Pay Encrypted Payload

In your request to the Google Pay API to obtain the encrypted payload be sure to include the following parameters:

- **type** - Smartpay only supports the *CARD* type.
- **allowedAuthMethods** - Smartpay can process both *PAN_ONLY* and *CRYPTOGRAM_3DS* authentication methods.
- **allowedCardNetworks** - specifiy the card networks that you wish to allow. If the customer has cards in their wallet that are not in the 'allowed' list then those cards will be greyed-out/disabled in their wallet.
- **merchantId** - found in the Google Pay Business Console under your account's *Public merchant profile* setting.
- **gateway** - a unique property that identifies Smartpay Advance as the processor; all encryption keys are associated with this ID. This field must be set to: `barclayssmartpayadvance`
- **gatewayMerchantId** - a unique property that identifies a Smartpay Advance or Bureau merchant. This field must be set to the *Enterprise ID* given to you by your Barclaycard account manager.

These fields can be seen in the following JSON example:

```
{
  'type': 'CARD',
  'parameters': {
    'allowedAuthMethods': [
      'PAN_ONLY',
      'CRYPTOGRAM_3DS'
    ],
    'allowedCardNetworks': [
      'AMEX',
      'MASTERCARD',
      'VISA'
    ]
  },
  'tokenizationSpecification': {
    'type': 'PAYMENT_GATEWAY',
    'parameters': {
      'gateway': 'barclayssmartpayadvance',
      'gatewayMerchantId': '<YOUR_SMARTPAY_ENTERPRISE_ID>'
    }
  }
}
```

3.4 Smartpay Payment Request

Having obtaining the encrypted payload from Google Pay you now need to send it to Smartpay in a Web Payment request. Smartpay will decrypt the payload, which contains a payment credential that's either a PAN or cryptogram, depending on how the card credentials are stored with Google Pay. Either cards are stored on file with Google (PAN_ONLY), and/or a device token on a device that's authenticated with a 3-D Secure cryptogram (CRYPTOGRAM_3DS).

In order to process a Google Pay transaction be sure to include the following fields in the webPaymentRequest, in addition to the standard payment request fields:

- paymentMethod = *GooglePay*
- authType = *AuthOnly* or *AuthAndSettle*
- storeResultPage = your merchant page that displays the transaction results
- card.encryptionSessionToken = encrypted payload received from Google Pay
- version = *30*, or above

Sample Web Payment Request (integrating directly with Googl Pay)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:msg="http://services.thelogicgroup.biz/EMIS_WEBPAYMENT_3_0">
<soapenv:Header/>
<soapenv:Body>
<msg:beginWebPayment>
<arg0>
    <requester>
        <authToken>A742DD37E4F7C903B95BAA847F034F65B85AC6739BCA9A4FD80A9B4E5AB8AB75</authToken>
        <enterpriseID>MERCHANT01</enterpriseID>
        <clientID>CLNT01</clientID>
        <transNo>123456</transNo>
        <environment>ECommerce</environment>
        <version>28</version>
    </requester>
    <transactionTime>2022-03-24T11:21:14.894</transactionTime>
    <acquirerReferenceData></acquirerReferenceData>
    <authType>AuthAndSettle</authType>
    <authenticate>true</authenticate>
    <billingAddress>
        <city>Fleet</city>
```

```
        <country>GBR</country>
        <county>Hampshire</county>
        <firstName></firstName>
        <lastName></lastName>
        <line1>123 High Street</line1>
        <line2></line2>
        <line3></line3>
        <middleName></middleName>
        <name>Billing Address</name>
        <postcode>GU51 3SB</postcode>
        <stateCode></stateCode>
    </billingAddress>
    <billingEmail>billing@email.com</billingEmail>
    <card>

<encryptionSessionToken>{"signature":"MEUCIQC9z5vgyTuGq9EEzMkXb+SDahEf/IrcEFt0adND4WlN4QIgB0wdbeEYnIld0iavV8SWjk
DFGRy2X6ML/iTQdWNkKRk\u003d","intermediateSigningKey":{"signedKey":"{\"keyValue\":\"MFkwEwYHKoZIzj0CAQYIKoZIzj0D
AQcDQgAE77m3kK3OPcm/oCdveKcE9AhzSC+YuBQMCT80FQ/MB99Eh7Ope+4eBHKWYugLERQJ8RWnzo8qYzOu+87J9jq5Bw\\u003d\\u003d\",\
"keyExpiration\":\"1648784148503\"}","signatures":["MEUCIBknp3rAcnZqzHvAhpXX2stpyW5uCPVhXV6U/7AXXSKEAiEAnKT9BS+j
1cJMZnwaPYKjOmgSv045FDRguqqB4LrCLFU\u003d"]},"protocolVersion":"ECv2","signedMessage":"{\"encryptedMessage\":\"V
CyI+RAF2mG14DZgyexA9aN5E7xRWh1MEPn4lAT2qsURza204hZ3cVjRBSoWTZs6yRUGlxgumUV/CI6aeV4wB3h60ItzRx00unXO+4ayPFpk9bHsO
frJUz4KV080azUuX84byQ0LF198784G9eWIkn1M6iFLL0ZvaL1hLfSEns7RhObTv9Fv+99xyHVTN3Hjoi0umOSqixwzroL1kjWG/Po7fb19dQH1H
Gb6b8NTHjXVIjeZqh8T1rfnfXIJjbJpGI4cp1GWzlfgG77nz9TD0GQ5KpOrj61fM+bv+JC9WBu9KLHCg/unXcmxU47XQTKoScHKzlZggVayEmWHH
9GshRdxpdWRZ2D9TGNK0N3fIOix4InWfIEgn7tQ5I55QiScfUTrfBO7/ANR98VxT87kfjCWlZZOvOjZb7nFnS19gUsg8YFcYojPaSWV1+dSJx5Hc
L7Wu28b9F351Wqj/pH+GBcWTzXpi1jROjshLxA4P778rUaZ4dFP32vqt1MttOYjTxi8C7yDJLZTLgoPWkIPf4RBCkhB4WUm9sbng33QblCC5Q+xH
mmTvWJ0lnhSJeFwKxP/2/sTAg\\u003d\\u003d\",\"ephemeralPublicKey\":\"BEcYTfI5vcLj9lP4LlXG/Z1qSyYdXUzFN0f9AaW3NeHKn
SjrLqxXEWKuQkfPsRvpp0tOECGGj3fVczW48k/HwMY\\u003d\",\"tag\":\"iGo2HBPL8OyiYk1zaMTxhLVe5CbMBsyuvTs2pBamdYk\\u003d
\"}"}</encryptionSessionToken>
    </card>
    <currencyCode>GBP</currencyCode>
    <customerID></customerID>
    <dcc checkDCC="false" performDCC="false"/>
    <fraudProfile></fraudProfile>
    <fraudProvider></fraudProvider>
    <fraudType>BeforeAuthorisation</fraudType>
    <invoiceID></invoiceID>
    <merchantTransactionID>0009</merchantTransactionID>
    <operatorID></operatorID>
    <paymentMethod>GooglePay</paymentMethod>
    <purchaseAmount>24999</purchaseAmount>
    <purchaseDescription>Samsung Galaxy S7 Edge</purchaseDescription>
    <salesDetails>Testing</salesDetails>
    <schemeReferenceData></schemeReferenceData>
    <shippingAddress>
        <city>Fleet</city>
        <country>GBR</country>
        <county>Hampshire</county>
        <firstName></firstName>
        <lastName></lastName>
        <line1>123 High Street</line1>
        <line2></line2>
        <line3></line3>
        <middleName></middleName>
        <name>Shipping Address</name>
        <postcode>GU51 3SB</postcode>
        <stateCode></stateCode>
    </shippingAddress>
    <storeResultPage> http://localhost:8080/DemoShop/response.jsf </storeResultPage>
    <styleSheetID></styleSheetID>
    <validate>false</validate>
</arg0></msg:beginWebPayment>
</soapenv:Body>
</soapenv:Envelope>
```


Note how the encrypted payload (encryptionSessionToken) appears within the request (Base64 text). After receiving the payload Smartpay will decrypt it using Google keys to extract the payment credential, which will be either a PAN or 3DS cryptogram, depending on the Google Pay authMethod. Please note that Smartpay uses the PAN for the current transaction only and doesn't store it wi thin its database. Here's examples of what the payload might contain as seen by Smartpay once decrypted:

PAN Only example

The following code snippet shows the JSON payload, decrypted by Smartpay, for a PAN_ONLY authMethod:

```
{
  "paymentMethod": "CARD",
  "paymentMethodDetails": {
    "authMethod": "PAN_ONLY",
    "pan": "4444333322221111",
    "expirationMonth": 10,
    "expirationYear": 2025,
    "assuranceDetails": {
      "accountVerified": true,
      "cardHolderAuthenticated": false
    }
  },
  "gatewayMerchantId": "some-merchant-id",
  "messageId": "some-message-id",
  "messageExpiration": "1577862000000"
}
```

Cryptogram 3DS example

The following code snippet shows the JSON payload, decrypted by Smartpay, for a CRYPTOGRAM_3DS authMethod. Note the additional fields for 'cryptogram' and 'eciIndicator' (the 3DS authentication flag that may be returned for tokens on the Visa card network).

```
{
  "paymentMethod": "CARD",
  "paymentMethodDetails": {
    "authMethod": "CRYPTOGRAM_3DS",
    "pan": "4444333322221111",
    "expirationMonth": 10,
    "expirationYear": 2025,
    "cryptogram": "AAAAAA...",
    "eciIndicator": "eci indicator"
      "assuranceDetails": {
      "accountVerified": true,
      "cardHolderAuthenticated": true
    }
  },

  "gatewayMerchantId": "some-merchant-id",
  "messageId": "some-message-id",
  "messageExpiration": "1577862000000"
}
```

3.5 PAN vs Cryptogram Flows

3.5.1 PAN Payments

If the payment credential is a PAN then the payment flow will automatically follow the standard 3-D Secure authentication and authorisation flow (unless it's subject to TRA exemption).

3.5.2 3DS Cryptogram Payments

If the payment credential is a cryptogram then this is exempt from 3-D Secure authentication and so the transaction will proceed directly to authorisation. Authentication has already been undertaken on the device and so having the token cryptogram and correct ECI meets PSD2 SCA requirements and provides liability shift to protect the merchant against fraud. The transaction status flow for Google Pay payments follows the basic card payment flow, apart from a couple of differences depending on the contents of the payload. Fraud checking via Kount or ACI, and CV2/AVS checking, may be performed only when the payment credential in the decrypted payload is a PAN; otherwise for a 3DS cryptogram these stages will be skipped.

When an intermediate response is received the transaction status will determine the next action to be taken. In some cases this will require you to redirect the customer's browser to a URL supplied in the response, or it may require you to decide whether to continue

with the transaction, by submitting an updateWebPayment call; alternatively, you may decide to abort the transaction by submitting a cancelWebPayment call.

3.6 Smartpay Payment Response

Sample Web Payment Response for Google Pay

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<ns2:beginWebPaymentResponse xmlns:ns2="http://services.thelogicgroup.biz/EMIS_WEBPAYMENT_3_0">
    <return>
        <acquirer>
            <ID>V</ID>
            <name>VISA</name>
        </acquirer>
        <authType>AuthAndSettle</authType>
        <authenticationRequest>
        <threeDSMethodNotificationURL> http://localhost:8080/DemoShop/acsNotification.jsf
</threeDSMethodNotificationURL>
        </authenticationRequest>
        <authenticationResponse>
            <cardScheme>VISA</cardScheme>
            <cavv/>
            <cavvAlgorithm>NOT_SET</cavvAlgorithm>
            <eci/>
            <errorCode>0</errorCode>
            <errorMessage>Card Not Enrolled</errorMessage>
            <invalidRequestDetail/>
            <invalidRequestErrorCode/>
            <invalidRequestVendorCode/>
            <merchantID>3501</merchantID>
            <status>NOT_ENROLLED</status>
            <threeDSVersionNumber>UNKNOWN</threeDSVersionNumber>
            <xid>9IIYk/3Xhaql10P5qYXHmHT8Tgs=</xid>
        </authenticationResponse>
        <authorisationResponse>
            <authcode>D12345</authcode>
            <hostResponseCode>00</hostResponseCode>
            <hostResponseMessage>Approved</hostResponseMessage>
            <merchantNo>1211118</merchantNo>
            <paymentMethod>GooglePay</paymentMethod>
            <softDeclined>false</softDeclined>
            <tid>23368318</tid>
        </authorisationResponse>
        <bank>
            <ID>B</ID>
            <name>Barclays-Dummy-0001</name>
        </bank>
        <billingAddress>
            <city>Fleet</city>
            <country>GBR</country>
            <county>Hampshire</county>
            <firstName/>
            <lastName>Billing Address</lastName>
            <line1>123 High Street</line1>
            <line2/>
            <line3/>
            <name>Billing Address</name>
            <postcode>GU51 3SB</postcode>
            <stateCode/>
        </billingAddress>
        <billingEmail>billing@email.com</billingEmail>
        <captures>
            <capture id="1">
                <amount>24999</amount>
                <outcome>OK</outcome>
            </capture>
```

```
        </captures>
        <card>
            <source>keyed</source>
            <maskedPAN>444433******1111</maskedPAN>
            <onlineToken>4444333322221111</onlineToken>
            <updateOnlineToken>true</updateOnlineToken>
            <onlineTokenUpdated>false</onlineTokenUpdated>
            <expiry>2027-12</expiry>
            <wallet>
                <type>GooglePay</type>
                <id>GP</id>
            </wallet>
            <cardType>V</cardType>
        </card>
        <configParams>
            <displayBasketDetails>false</displayBasketDetails>
            <merchantStates/>
            <storeID>100001</storeID>
        </configParams>
        <currencyCode>GBP</currencyCode>
        <customerID/>
        <cvResponse>
            <rawCv2>no_information</rawCv2>
            <rawAddress>no_information</rawAddress>
            <rawPostcode>no_information</rawPostcode>
            <result>NoDecision</result>
        </cvResponse>
        <environment>ECommerce</environment>
        <merchantTransactionID>0009</merchantTransactionID>
        <purchaseAmount>24999</purchaseAmount>
        <purchaseDescription>Samsung Galaxy S7 Edge</purchaseDescription>
        <requester>
            <enterpriseID>MERCHANT01</enterpriseID>
            <clientID>CLNT01</clientID>
            <transNo>123456</transNo>
            <version>28</version>
        </requester>
        <responder>
            <name>EMIS-AG</name>
            <version>3.55.2.0 Build:c8eaf03b32dc.PROD</version>
            <releaseDate>09/02/2022</releaseDate>
            <id>TSI-COR301-P1</id>
        </responder>
        <response>validated</response>
        <salesDetails>Testing</salesDetails>
        <schemeReferenceData>01000000003755712AB</schemeReferenceData>
        <shippingAddress>
            <city>Fleet</city>
            <country>GBR</country>
            <county>Hampshire</county>
            <firstName/>
            <lastName>Shipping Address</lastName>
            <line1>123 High Street</line1>
            <line2/>
            <line3/>
            <name>Shipping Address</name>
            <postcode>GU51 3SB</postcode>
            <stateCode/>
        </shippingAddress>
        <status>CAPTURED</status>
        <storeResultPage> http://localhost:8080/DemoShop/response.jsf </storeResultPage>
        <styleSheetID/>
        <totalAmount>24999</totalAmount>
        <transactionReference>bd88eefd-94d6-4850-a570-8c7bf4329e1b</transactionReference>
        <validCardTypes>Amex</validCardTypes>
        <validCardTypes>Visa Business Debit</validCardTypes>
        <validCardTypes>Visa Debit</validCardTypes>
        <validCardTypes>Visa Electron Prepaid</validCardTypes>
        <validCardTypes>VISA</validCardTypes>
```

```
        <validCardTypes>Mastercard Commercial Prepaid</validCardTypes>
        <validCardTypes>Visa Business Prepaid</validCardTypes>
        <validCardTypes>International Maestro</validCardTypes>
        <validCardTypes>Visa Electron</validCardTypes>
        <validCardTypes>Mastercard Commercial Prepaid Credit</validCardTypes>
        <validCardTypes>Mastercard Prepaid</validCardTypes>
        <validCardTypes>Visa Prepaid</validCardTypes>
        <validCardTypes>Mastercard Debit</validCardTypes>
        <validCardTypes>Mastercard Commercial Debit</validCardTypes>
        <validCardTypes>Mastercard</validCardTypes>
        <validPayment>GooglePay</validPayment>
    </return>
</ns2:beginWebPaymentResponse>
</soap:Body>
</soap:Envelope>
```

4 Payment Choice State

If you initiated a web payment request without specifying a paymentMethod, so that it's currently in the *PAYMENT_CHOICE* state, and *GooglePay* is listed as an available validPayment in the response, then you can submit an updateWebPayment request with the paymentMethod set to *GooglePay* in order to provide card details.

5 Enabling Google Pay in Smartpay

Before you can start accepting payments from Google Pay using Smartpay you'll need to request your Barclaycard account manager to enable Google Pay within your merchant's configuration, if not done so already; they will also manage all required certificates and keys with Google Pay.

What's New

API version 30 (Smartpay Core release 3.56.1.x) 12 April 2022 (*Document version 02*)

- Added support for integration using Hosted Payment Page

API version 29 (Smartpay Core release 3.55.2.x) 11 March 2022 (*Document version 01*)

- Added Google Pay digital wallet payments to Web Payments SOAP API