



## Google Pay™ integration

---

Version 1.0

# Changes to the document

| <b>Date</b> | <b>Changing</b> | <b>Version</b> |
|-------------|-----------------|----------------|
| 2024-05-21  | Initial version | 1.0            |

|   |          |
|---|----------|
| <b>Changes to the document</b> .....  | <b>2</b> |
| <b>Introduction</b> .....   | <b>4</b> |
| <b>Google Pay &amp; Corvus Pay</b> .....  | <b>4</b> |
| Using the CorvusPay payment method (standard <i>redirect</i> integration) ..... | 4        |
| Using the CorvusPay API .....   | 5        |
| Google Pay API Web Integration .....  | 5        |
| Google Pay API Android Integration .....  | 5        |
| Google Pay API configuration .....  | 5        |
| Payment processing .....  | 6        |
| Secure Customer Authentication (SCA/3DS) and PSD2 .....                         | 7        |

# Introduction

**Google Pay™ is a digital wallet for online payments offered by Google.**

It is an advanced and secure solution that customers can use to add credit/debit cards to their wallet and use them to pay at webshops or mobile applications that use the CorvusPay service.

By combining the Google API with the CorvusPay API, the payment process is fast, easy and secure.

Increase your conversion rate with these features:

- Your customers simply select a card from their Google Pay wallet.
- Google Pay and CorvusPay manage all sensitive user data for you.
- A secure encryption/decryption method raises your customers' trust in the entire payment process and significantly reduces online fraud.

To make all this possible, Google Pay tokenizes your customers' cards as follows:

- Tokenized cards outside of card schemes: standard physical credit cards that require 3D-Secure verification for additional fraud protection. This is only applicable for payments through your point of sale, and this method of authorization is called PAN\_ONLY.
- Cards tokenized within card schemes: A virtual card with a device-specific account number (DPAN). These types of cards are used in place of a real plastic card. Since the digital signature of the mobile device used for payment protects these cards from fraud, 3D-Secure is unnecessary. This only applies to users who use Google Digital Wallet on their mobile device, and the authorization method is called CRYPTOGRAM\_3DS.

To pay with Google Pay, your customers can use cards of the following brands: Visa, MasterCard, American Express, Discover, JCB.

## Google Pay & CorvusPay

To use the Google Pay service, it is necessary to integrate with the CorvusPay system in one of two ways:

- Standard *redirect* integration using CorvusPay payment form
- Integration via CorvusPay API

To activate the Google Pay payment method through the CorvusPay system, you need to do the following:

- Submit a request to our customer service [support@CorvusPay.com](mailto:support@CorvusPay.com)
- Follow the Google Pay API Terms of Use <https://payments.developers.google.com/terms/aup>

### Using the CorvusPay payment method (standard *redirect* integration)

Before activating the Google Pay payment method on the CorvusPay payment form, the merchant must complete the integration with the CorvusPay system.

After activating the Google Pay payment method, a Google Pay tab will be added to the CorvusPay payment form, which will enable Google Pay payment.

## Using the CorvusPay API

CorvusPay supports Google Pay transactions directly through its API. CorvusPay provides CRYPTOGRAM\_3DS authorization method for Google Pay.

The integration itself includes the following steps:

1. Merchants integrate with the Google Pay API.
2. The customer selects the Google Pay button on the webshop or the merchant's mobile app.
3. The customer confirms the payment and Google Pay returns the payment token to the merchant.
4. The merchant encodes the received token using Base64 and sends it to the Corvus Pay API as part of the POST request at /api/1.6/auth.

Before you start with integration to Google Pay API you must accept the Google Pay API Terms of Service <https://payments.developers.google.com/terms/sellertos>

### *Google Pay API Web Integration*

To integrate your webshop with the Google Pay API, follow the instructions provided in the [Google Pay API documentation for web applications](#). Familiarize yourself with the [Google Pay web application brand guidelines](#) and see the Google Pay integration guidelines to [Google Pay web application integration checklist](#).

### *Google Pay API Android Integration*

To integrate your Android app with the Google Pay API, follow the instructions provided in the [Google Pay documentation for Android developers](#). Familiarize yourself with the [Google Pay Android App Brand Guidelines](#) and see the [Google Pay Android Integration Checklist guidelines](#).

### *Google Pay API configuration*

To initiate a transaction, your web or mobile app must send a request to the Google Pay API, specifying the payment method within the [PaymentMethod](#) object.

Configuration parameters

Set the following properties under "parameters":

*"allowedAuthMethods": [ "CRYPTOGRAM\_3DS" ]*

*"allowedCardNetworks": [ "AMEX", "DISCOVER", "JCB", "MASTERCARD", "VISA" ]*

Set the following properties under "tokenizationSpecification.parameters":

- "gateway": "corvuspay"
- "gatewayMerchantId": Use your storeID.

Example of a PaymentMethod object:

```
{
  "type": "CARD",
  "parameters": {
    "allowedAuthMethods": [ "CRYPTOGRAM_3DS" ],
    "allowedCardNetworks": [ "AMEX", "DISCOVER", "JCB", "MASTERCARD", "VISA" ]
  },
}
```

```
    "tokenizationSpecification": {
      "type": "PAYMENT_GATEWAY",
      "parameters": {
        "gateway": "corvuspay",
        "gatewayMerchantId": "602"
      }
    }
  }
}
```

### Payment processing

After successfully initiating a transaction via the Google Pay API on your web or mobile application, you will receive a `PaymentData` object in the response. To complete the payment, it is necessary to pass the `"token"` parameter (`paymentData.paymentMethodData.tokenizationData.token`) to the CorvusPay API. The parameter needs to be encoded using the Base64 algorithm before being sent. This encoded value should be sent as the value of the `"thirdPartyTokenData"` parameter, along with `"thirdPartyTokenType"` set to `"GOOGLE_PAY"`, and all other necessary parameters according to the integration documentation on the `"/api/1.6/auth"` endpoint.

Example:

```
POST /api/1.6/auth
Headers:
Content-Type: application/json
Accept: application/json

Body:
{
  "storeID": "602",
  "orderNumber": " e5b913",
  "language": "hr",
  "amount": "100.23",
  "currency": "EUR",
  "cart": "Shoes",
  "cardholderName": "Pero",
  "cardholderSurname": "Peric",
  "cardholderEmail": "peroperic@gmail.com",
  "cardholderIp": "234.234.234.432",
  "cardholderCountryCode": "HR",
  "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 Safari/537.36",
  "thirdPartyTokenType": "GOOGLE_PAY",
  "thirdPartyTokenData":
"eyJzaWduYXR1cmUiOiJNRVlDSVFEZi85QXc4NE41VXBWFwMGTUStRlUptUDRGbTRMYSticE9udm1aZXgyd0lo
QUtBM1Q3UUXKNDRuNHNqbK5Yb3ZkdCtVz2VXTUZIR2N0TWFQOGtPcS93czUiLCJpbmRlcm1lZG1hdGVtaWduaW5nS
2V5Ijpw7InNpZ25lZEtleSI6IntcImtleVZhbHVlXCI6XCJNRmt3RXdZSEtVWk16ajBDQVFZSUtvWk16ajBEQVFjRF
FnQUVpWGY2bkR5eXYxL2lQcThZa1prMGJ5OHhwM1ZteGpGVtNtOVhZOWhxL3N5cG1VYjRaYVhBNTN1dnNNVEJnS09
aNVJSL1hURHRzQ0NldlFmRE10RWFzZ1xcdTAWM2RcXHUwMDNkXCIsXCJrZX1FeHBpcmF0aW9uXCi6XCiXNzE1NjU0
NDA1NzM2XCJ9Iiwic2lnbmF0dXJlcyI6WyJNRVFDZSUNS3ZnT1ZYZ0JuvGt1M1IvOU9Fd3BMNHZmSjh3Rzd1NkZMV
E93L3c4My9BaUJlQmd1Y09oVTZ0K2p4S0l6Tm9IRWeySGdXOVNiM1h1S1ZHdi9oeW52b2VRXHUwMDNkXHUwMDNkI1
19LCJwcm90b2NvbFZlcnNpb24iOiJFQ3YyIiwic2lnbmVkdGVzZ2FnZSI6IntcImVucyY3J5cHRlZE1lc3NhZ2VcIjpw
cIjdBTRWxuY2NBb3AvSSszU3FqOUJ6d0ZNVnB6dXZEn1YzktKbUVwaXZsNUUpGYmh2eW5QRWx0Q21obEUvSm80MVNn
T1JRQ250N2xPdFY4bjZBaW84OU1yU21SejBlRFRqZTZHaWx3aFNYSgdcml2cmNDd2RBNGp3SDRsV1pHWDJKwFJqL
ytITWp1WmxhQkYxZDhLVGxTYTY2TURlbnMjISHg3YVRZ0Gx6Y1BXUTN5b2JUVHplRzJcWjAyRzRXNXhwVbWScw1GL0
Y1dGJXeXhpMUFGVkMvMjFRL1pCckN1MH14V09hcTRsWTJzRj1zNnFrODFqZ312b09WskQ0UGpQRHdRd3pMbGdxV25
ONE1Va1M1ZjQxcXdqSjZrOWZSWDB0L0FvYko2YnhNRkNtL05qREY1Yy9vRzNhQ1ZlRDJEUk9hdytpT01Nc2xnQkNW
bFNoMk91RW82RzcrQ0ZzSDRWSV1tWE9xRW8wYXhDdkdVU3YrcXZ0SFV1TGRoMWhoOWZvcnVWdk1udVY2Nm1HWThNe
jZaSVdGw1lpNklzV1JWt3Yxa01DODVhSHE1QThHVMeZTnoxOEpCQk8wZ1B5WUtxdzdYzWmzdmUL0Rvb0RmMndQTi
svOEPhL3VONkYJVVMyQ0x3QjFZV1RXtmIwV31Ebmh3ZmNmZRFV2ZvNjVuem1pcDRNTHJnMnF4eVR5dw5tYwXTZUD
aUw1UL3DNbkdyT0t6ckZkbDhnXfx1MDAzZFwiLFwiZXB0ZW11cmFsuHVibG1jS2V5XCi6XCiSkpXTkrnZi9TSzBJ
NzYzdj12dFdRdTFyV3htVloyY3VWODRzMC9PdHYvTwt1eUtyOXNPbXZqdFRBTGVOM1NvdGFzNmNKbVJIdW51N1JaM
```

```
TRNU2pKw1FcXHUwMDNkXCIsXCJ0YwdcIjpcIitwTctjUUMyL1h1cDYvWU9vQURBbk4ybw52SUNRV3ZiUEJueWNSaU
FseVlcXHUwMDNkXCJ9In0=",
  "requireComplete": false,
  "signature": "070168fe121daa434a98ec33a29904d6f3a1cfac561232ed1414869f72529b2a"
}
```

Note: In this case, the parameters "cardNumber", "ccMonth" and "ccYear" should be omitted.

#### *Secure Customer Authentication (SCA/3DS) and PSD2*

Google Pay supports CRYPTOGRAM\_3DS authentication method that is SCA (Strong Customer Authentication) compliant, thus avoiding the need for additional 3DS verification.