



Planet Pay sp. z o.o.
ul. Jutrzenki 139, 02-231 Warszawa
www.planetpay.pl

Sąd Rejonowy dla m. st. Warszawy XIV Wydział Gospodarczy
KRS 0000427567, NIP 679-308-33-13, REGON 122603652

Google Pay

Documentation

Version 1.0



TABLE OF CONTENTS

1. INTRODUCTION	3
2. INTEGRATION	3
2.1. INTEGRATION VIA WEBSITE	3
2.2. ANDROID INTEGRATION	4
3. PAYMENT INITIATION	4



1. INTRODUCTION

Google Pay™ is a fast and easy way to pay in millions of places: the Internet, shops and many more. It offers everything that is needed for payment, and secures the data (the card data are safely stored by Google). This payment method can be used with all devices (phones and computers), regardless of the operating system or browser used.

If the Merchant uses Google Pay, they undertake to observe the provisions of the service terms and conditions available at:

- <https://payments.developers.google.com/terms/sellertos>

2. INTEGRATION

2.1. INTEGRATION VIA WEBSITE

This process is best described in the documentation of Google Pay itself available at:

- <https://developers.google.com/pay/api/web>

and particularly the tutorial that can be found here:

- <https://developers.google.com/pay/api/web/guides/tutorial>

When implementing the interface, it is advisable to follow the checklist available here:

- <https://developers.google.com/pay/api/web/guides/test-and-deploy/integration-checklist>

as well as the recommendations and guidelines from Google:

- <https://developers.google.com/pay/api/web/guides/brand-guidelines>

The gateway parameter should be **itcardpaymentservice**:

```
const tokenizationSpecification = {
  type: 'PAYMENT_GATEWAY',
  parameters: {
    'gateway': 'itcardpaymentservice',
    'gatewayMerchantId': MID
  }
}
```

whereas the merchant's identifier assigned by ITCARD should be given as **MID**.

eCommerce API supports Visa and Mastercard cards, so the list of permitted card networks should be as follows:

```
const allowedCardNetworks = ["MASTERCARD", "VISA"];
```



A token generated by GPay should be sent to the eCommerce API without any modifications.

The eCommerce API is described in the document enclosed below:



eCommerce API -
technical specification

2.2. ANDROID INTEGRATION

The process of Android integration is best described in the documentation of Google Pay itself available at:

- <https://developers.google.com/pay/api/android>

When implementing the interface, it is advisable to follow the checklist available here:

- <https://developers.google.com/pay/api/android/guides/test-and-deploy/integration-checklist>

as well as the recommendations and guidelines from Google:

- <https://developers.google.com/pay/api/android/guides/brand-guidelines>

For integration purposes, you should use the SDK API presented in the document enclosed below:



SDK Payment.pdf

3. PAYMENT INITIATION

In order to be able to pay with a card saved at Google Pay, you should follow the below scheme.

The data generated from the GPAY component are sent to the payment gateway with the call:

```
PUT /payment/{paymentId}/instrument
{
  instrument: {
    type: "GPAY",
    token: "tokenbase64encoded"
  }
}
```



Alternatively (if it is required to send data of the device from which the client performs the operation):

```
PATCH /payment/{paymentId}

{
  instrument: {
    type: "GPAY",
    token: "tokenbase64encoded"
  }
  device: {
    ip: "172.12.12.12"
  }
}
```

The payment is initialized earlier in a standard way with the call of POST /payment (as described in the ecommerce REST API).

If it is necessary to conduct SCA (strong customer authentication) with the use of the 3DSecure standard, relevant data will be sent in reply.

```
{
  "paymentId": "3JrHowwAmM01b1DqNmUq0F",
  "status": "PENDING",
  "dataRequest": "CARD_3DS",
  "acsData": {
    "acsURL": "acsserveraddress",
    "creqPayload": "acsdatatosend"
  }
}
```

When the transaction is completed, the merchant should ask for the status or integrate with a dedicated notification channel (as described in the specification for the ecommerce REST API).

All requests must include the authentication token in the JWT format:

```
Authorization: Bearer {access_token}
```