

# Google Pay™ Online Payments

## HOSTED PAYMENT PAGE SOLUTION TECHNICAL DOCUMENTATION




This part discusses procedure for Google Pay payment method integration. The rest of the protocol does not change.

Google Pay payments can only be used for Purchase type authorizations.

Card networks accepted: Visa, Mastercard. 3DS for Google Pay is mandatory and enabled for all merchants.

Merchant redirects users to TBC's hosted checkout page where final interaction with Google Pay button is done and payment is processed.

Merchant is mandated to display Google Pay button according to Google guidelines.

Payment type	Payment flow	Countries	Currencies	Recurring	Refund/Partial Refunds	Reversal/Partial reversal
Wallet	Direct	International	Multiple			

# GOOGLE PAY™ FOR HOSTED PAYMENT PAGE

## BEFORE YOU BEGIN

These instructions explain how to add Google Pay to your existing API-only integration. The API-only integration works the same way for all payment methods. If you haven't done this integration yet, refer to our [Overview \(tbcbank.ge\)](#)

Make sure that you are familiar with Google Pay APIs [Acceptable Use Policy](#) and accept the terms defined in the [Google Pay API Terms of Service](#) . Please note, that the billing address is not required, as it is not requested by either Visa/MC domestically. You will get the payment indicators:

gateway - UNITEDFINANCIALCORPORATION

gatewayMerchantID - THE GATEWAY MERCHANT-ID WILL BE GIVEN TO YOU DIRECTLY WHEN ONBOARDING.

merchant initiates payment request just after, that the check and all the pre-works are done on behalf of merchant and the payment is ready to proceed.

## MAKE A PAYMENT

1. Get [access token](#)
2. From your server [create a payment](#)
  - Payment method: Google pay
  - Method Code: **14**
  - returnUrl: Url to redirect user after finishing payment
    - Example: shopping.ge/callback
  - callbackUrl: when payment status changes to final status, POST request containing PaymentId in the body will be sent to given URL. In case of successful receipt of the request, the merchant must return the status code 200 and check the payment status through GET /payments/{payment-id} endpoint

## REQUEST SAMPLE

```
{
  "amount": {
    "currency": "GEL",
    "total": 200,
    "subTotal": 0,
    "tax": 0,
    "shipping": 0
  },
  "returnurl": "test.ge/callback",
  "extra": "GE60TB4572261006330008",
  "userIpAddress": "127.0.0.1",
  "expirationMinutes": "5",
}
```

```

"methods" : [5, 7, 14],
"installmentProducts":
[
  {"Name":"t1","Price":100,"Quantity":1},
  {"Name":"t1","Price":50,"Quantity":1},
  {"Name":"t1","Price":50,"Quantity":1}
],
"callbackUrl":"https://test.ge",
"preAuth":false,
"language":"EN",
"merchantPaymentId": "P123123",
"saveCard": true,
"saveCardToDate": "1021"
}

```

## RESPONSE SAMPLE

```

"payId": "tpay-tbvqma2372015",
"status": "Created",
"currency": "GEL",
"amount": 200,
"links": [
  {
    "uri": "https://api.tbcbank.ge/v1/tpay/payments/tpay-tbvqma2372015",
    "method": "GET",
    "rel": "self"
  },
  {
    "uri": "https://tpay.tbcbank.ge/checkout/choose-payment-method/tpay-tbvqma2372015",
    "method": "REDIRECT",
    "rel": "approval_url"
  }
],
"transactionId": null,
"preAuth": false,
"recId": "8715aBbB4314e7dC4f91482283988365a3c30822",
"httpStatusCode": 200,
"developerMessage": null,
"userMessage": null
}

```

## GET PAYMENT DETAILS

Request returns [payment status](#) and details for given payId

## REQUEST SAMPLE

```

curl --location --request GET 'https://api.tbcbank.ge/v1/tpay/payments/' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'apikey: 1XcdL8JJiAN8Vjlu6NW3kNece0Qo1wnF' \

```



## BRAND GUIDLINE

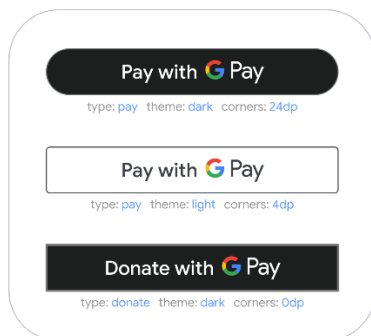
# GOOGLE PAY™ PAYMENTS BUTTONS

All Google Pay payment buttons displayed brand guidelines, which include, but aren't limited to the following:

- Size relative to other similar buttons or elements of the page
- Contrast with the background color of the area surrounding it
- Clear space

## CUSTOMIZE YOUR BUTTON

The new Google Pay button view lets you customize the button **theme**, **shape** and **corner roundness** to match your UI design.



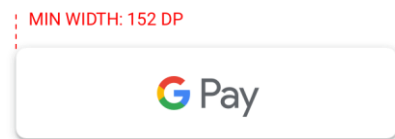
## CLEAR SPACE

Always maintain the minimum clear space of 8 dp on all sides of the payment button. Ensure the clear space is never broken with graphics or text.



## MINIMUM WIDTH

The Google Pay™ button should have a minimum width of 90 dp. All "Buy with Google Pay" payment buttons should have a minimum width of 152 dp.



Don't create buttons with your own localized text.

### Button color:

Default	Black	White

### Button type:

- Pay

### Button locale:

- English

**LIBRARY JAVASCRIPT:****Default:**

```

<!-- html fragment -->
<div id="container" style="width: 240px; height: 40px;"></div>

// javascript fragment
const container = document.getElementById('container');
const button = googlePayClient.createButton({
  buttonColor: 'default',
  buttonType: 'pay',
  buttonSizeMode: 'fill',
  onClick: () => {},
  allowedPaymentMethods: [] // use the same payment methods as for the loadPaymentData() API call
});

container.appendChild(button);

```

**Black:**

```

<!-- html fragment -->
<div id="container" style="width: 240px; height: 40px;"></div>

// javascript fragment
const container = document.getElementById('container');
const button = googlePayClient.createButton({
  buttonColor: 'black',
  buttonType: 'pay',
  buttonSizeMode: 'fill',
  onClick: () => {},
  allowedPaymentMethods: [] // use the same payment methods as for the loadPaymentData() API call
});

container.appendChild(button);

```

**White:**

```

<!-- html fragment -->
<div id="container" style="width: 240px; height: 40px;"></div>

// javascript fragment
const container = document.getElementById('container');
const button = googlePayClient.createButton({
  buttonColor: 'white',
  buttonType: 'pay',
  buttonSizeMode: 'fill',
  onClick: () => {},
  allowedPaymentMethods: [] // use the same payment methods as for the loadPaymentData() API call
});

container.appendChild(button);

```

## Body Parameters

Parameter	Type	Description
<a href="#">amount {} (required)</a>	object	transaction amount
<b>returnurl (required)</b>	string	url to redirect user after finishing payment
<b>extra</b>	string	additional parameter for merchant specific info (optional). only non-unicode (ANS) symbols allowed. max length 25. This parameter will appear in the account statement
<b>extra2</b>	string	additional parameter for merchant specific info (optional). only non-unicode (ANS) symbols allowed. max length 52.
<b>expirationMinutes</b>	number	payment initiation expiration time in minutes.
<b>methods</b>	array	There are several payment methods available to be passed when initiating TBC E-Commerce payment. Detailed list and description of available methods is described in the <a href="#">Classifications chapter/ Payment Methods</a>
<a href="#">installmentProducts []</a>	array	list of installment products. mandatory if installment is selected as payment method. Please note, sum of prices of installment products should be same as total amount.
<b>callbackUrl</b>	string	when payment status changes to final status, POST request containing PaymentId in the body will be sent to given URL. In case of successful receipt of the request, the merchant must return the status code 200 and check the payment status through GET /payments/{payment-id} endpoint. Body example {"PaymentId":"6azzo13uw5u2116332"} callbackUrl must be added on the merchant dashboard at ecom.tbcpayments.ge. If callbackUrl parameter is not provided, system automatically takes the basic callbackUrl value from the merchant dashboard. Please, verify that POST request is allowed at your callbackUrl from the following IP addresses: 193.104.20.44 193.104.20.45 185.52.80.44 185.52.80.45
<b>preAuth</b>	boolean	specify if preauthorization is needed for the transaction. if "true" is passed, amount will be blocked on the card and additional request should be executed by merchant to complete payment. To finalize authorization process, /v1/tpay/payments/:paymentId/completion endpoint should be used. Note, that Completion request can be initiated not later than 30 days from Preauthorization date. By default block is saved for 30 days, although some banks may have a different setting, so this setting depends on the card issuing bank (Issuer Bank).  The following values are allowed: true, false
<b>language</b>	string	default language for payment page  The following values are allowed: KA, EN



<b>merchantPaymentId</b>	string	Merchant-side payment identifier
<b>skipInfoMessage</b>	boolean	If true is passed, TBC E-Commerce info message will be skipped and customer will be redirected to merchant. If false is passed or this parameter isn't passed at all, TBC E-Commerce info message will be shown and customer will be redirected to merchant.
<b>saveCard</b>	boolean	Specify if saving card function is needed. This function should be enabled for the merchant by bank. If true is passed, recId parameter should be returned in response, through this parameter merchant can execute payment by saved card - POST /payments/execution. Zero amount is allowed for this function. If card saving function is requested with preauthorization parameter=true, saved card execution method will be activated after preauthorization completion. WebQR, ApplePay and installments pay methods are not allowed for saving card request
<b>saveCardToDate</b>	string	The date until the card will be saved can be passed in following format "MMYY". If the saveCardToDate is not provided or data provided by the merchant exceeds card expiry, the system will automatically assign the SaveCardToDate value that will be equal to card expiry. The actual card save date must be verified by getting payment status after transaction completed with GET /payments/{payment-id}
<b>description</b>	string	Payment short description for clients, max length 30. This parameter will appear on the checkout page.