ADDITION TO API GateLine Payment Gateway

Integration with Google Pay™

General information

Google Pay is the fast, simple way to pay with phone and speed through checkout within apps and websites. There is no maximum transaction limit when your customer use phone and card. Payment info is also protected with multiple layers of security so the customer can pay with peace of mind – all the time. Get what the Customer need with the simple press of Google Pay button on the website or in app.

Getting started with Google Pay using API GateLine is simple — get set up in a few days.

Google Pay terms of use

You can use Google Pay technology on websites and mobile applications subject to the following requirements:

- [Brand guidelines](#)
- [Google Pay API terms of service](#)
- [Google Pay APIs acceptable use policy](#)

1. Using Google Pay with GateLine Payment Gateway

For merchants, there are three scripts of using Google Pay with GateLine Payment Gateway:

- The Google Pay payment button is placed on the GateLine payment form.

- The Google Pay payment button is located on the merchant's website.

- The Google Pay payment button is located in the merchant's mobile application.

Generally, the payment flow using the Google Pay button looks as follows:

1. The customer taps the Google Pay payment button and sees a payment sheet with a list of supported payment methods.

2. The customer selects a payment method and Google Pay securely returns a payment token for that method.

3. Website or app submits the payment token, along with details about the purchase, to its backend and sends a request to the GateLine.

4. To execute the payment, the backend of GateLine processes the purchase and sends the payment data to Bank Acquirer.

5. Depending on the settings of your bank terminal, for a non-tokenized payment method, the Customer will need to pass 3D authorization*****.

* **-** 3DS verification is automatically activates on Gateway side. There is no need to take any additional steps to enable it.

Accepted payment systems:

- Visa
- Mastercard
- Mir

1.1. Using Google Pay on GateLine Payment Form

The merchant should make only a standard integration with GateLine Payment Gateway using GateLine API without registration in Google Pay.

For making the payment, the Customer is redirected to the Gateline Payment Form where the Customer can enter payment card data or simple taps a Google Pay payment button, that appears on this form. After the payment, the Customer will be informed of the result and returned back to the Merchant's website or app.

To use Google Pay on the Gateline checkout page, you just need to inform support about the need to connect Google Pay.
The Google Pay button will automatically appear on the checkout page in compatible browsers at [devices that support Google Pay](#).

1.2. Using Google Pay on a merchant website

For implementation Google Pay payment button on the website the merchant needs the following:

1. A Google Account

2. Check [Google Pay Web Brand Guidelines](#)
3. Website must have HTTPS and support TLS version 1.2.

4.   The site domain must be previously registered and verified with Google ([Google Pay API Production Access Enablement Request Form](#)).

5.   Adhere to [Google Pay API Acceptable Use Policy](#) and receive a Google merchant ID.

When placing a button on website, Merchant is encouraged to review the following documentation:

- [Overview](#)

- [Integration checklist](#)

On the merchant's payment page, you need to connect the Google Pay API library:

```
<script src="https://pay.google.com/gp/p/js/pay.js"></script>
```

When requesting the Google Pay API, use the following parameters:

```javascript
const tokenizationSpecification = {
    type: 'PAYMENT_GATEWAY',
    parameters: {
        'gateway': 'onelya',
        'gatewayMerchantId': 'YOUR_ONELYA_SITE_UID'
    }
};
```

JS code example:

```
// loading required libraries
<script src="/js/payment/googlepay.js" crossorigin="anonymous"></script>
<script src="https://pay.google.com/gp/p/js/pay.js" async ></script>

<div id="googlepay_container"></div>
 // GateLine client initialization
window.addEventListener('load', function(ev) {
   on_googlepay_load({

       // Google Pay identifier
       merchant_id:        'merchant_id',

       // Merchant identifier in GateLine Gateway
       gateway_merchant_id: 'onelya_site_uid',

       amount:             '12.12',

       // Merchant server URL. To which the Google Pay payment token will be
forwarded
       payment_url:        '/GooglePay',

       // additial parameters for payment_url
       payment_url_params: { order_id: '123' },

       // For real payment, the value is set to PRODUCTION or this parameter is not
set at all
       environment:        'TEST',


       // container for adding a Google Pay payment button
       container:          '#googlepay_container',

       // Additional method, for example, for adjusting styles
       on_button_show: function(result) {
          let btn = result.button;
       },
```

```
    // helper method, for example to display a progress bar
    show_progress: function(show) {
       let spinner = document.getElementById('spinner');
       if (show) spinner.style.display = 'block';
       else spinner.style.display = 'none';
    },

       // Method for completing actions on the site page after payment.
    payment_result: function(result) {
       if (result.redirect) {
          window.location = result.redirect;
       }
       else if (result.form_data) {
          submit_form(result.form_data);
       }
       else if (result.error) {
           process_error();
       }
    }
  });
});
```

Calling this function performs all the actions that are neded to interact with Google Pay:

- Initializes PaymentAPI

- Assigns a handler to the payment button

- Makes PaymentRequest with all required parameters

- Send the result to the Merchant's server

After successful upload of Google Pay payment data, the payment token will be sent to the merchant's server at the url specified in payment_url.

Merchant's server request format:

```
{
    payload: 'AAasaadcefvASGFVvewfvw…',
    Merchant_specific_param: ',,,,'
```

}
payload - GooglePay payment token.

Sending the encrypted token to the GateLine server.

    After receiving the encrypted data, the Merchant's server resend it to the GateLine server as it is, using the **GateLine API method /order/googlepay** (point3)

Note:
It means that Merchant has entered into an agreement with GateLine to use the GateLine API.
Interaction by API is carried out via an encrypted SSL channel using certificates for this merchant

API method /order/googlepay example:

```
  my $api = GateLineAPI->new(
      key => 'merchant_secret_key',
      cert => 'merchant_cert`
);
$api->google_pay_request(payload => $payload, order_id => '12345');
```


Note:
For PAN_ONLY transactions, To complete 3-D Secure authentication, you need to divert a payer to the address specified in the AcsUrl parameter of Gateline server's response

See also:

- Google Pay API [Documentation](#) for Web.

- Google Pay API Web [Tutorial](#)

1.3. Using the Google Pay service in the Merchant's mobile application

This case is similar to the case with the placement of the Google Pay payment button on the Merchant's website (section 1.2.), except that the integration Merchant mobile application with Google Pay API is made in accordance [Google Pay recommendation](#).

Use the [Google Pay API](#) to display Google payment button in your application, receive payment data and send it to the GateLine Payment Gateway in accordance with the recommendations given in section 1.2.

See also:

- [Example](#) of using Google Pay API by Google
- [Google Pay Android brand guidelines.](#)
- [Google Pay Android integration checklist](#)

2. Method POST /order/googlepay

Accepted parameters:

Value name	Description	Example
merchant_order_id	Order ID in the client system	123456789
amount	Payment amount	100.99
payload**	paymentMethodData.tokenizationData.token, received from google pay. Must be encoded into base64	WdodCI6IjYyNSIsIldpbmRvd1dppZHRoIjoiMTM0MyJ9fQ==

extended.client_browser_info**	The client's browser data block is used in the terminals of PJSC VTB Bank.	See GateLine API Manual. paragraph 3.3.6
extended. cardnetwork**	PaymentMethodData.info .cardNetwork, received from google pay	See GateLine API Manual. paragraph 3.1.3.5
country *	country (ISO 3166-1 alpha-2)	US
state *	State (ISO 3166-2)	TX
city *	City	Springfield
postal_code *	Zip code	12345
address *	Customer address	123 Main Street
phone *	Customer phone(up to 20 symbols)	+12341231212
email *	E-mail	mail@example.com
Ip*	Customer IP-address	123.123.123.123
description *	Payment description	e Goods (order #123456789)
return_success_url *	Customers redirect URL in case of successful Transaction, according to the standard RFC 3986	http://example.com/success?
return_failure_url *	Customers redirect URL in case of failure Transaction, according to the standard RFC 3986	http://example.com/failure?
extended *	Extended parameters	See GateLine API Manual. paragraph 3.1.3.5

* – optional field

** – when using the script provided by the Gateline support(paragraph 2.2), payload and client_browser_info parameters are generated automatically in the required encoding

Response fields:

Имя	Описание	Пример
order_id	Order ID in the Gateline system	689b987e82aa17d5151807781 360baf9
merchant_order_id	Order ID in the client system	123456789
status	Transaction status	success
message	Result description	Success
descriptor	Terminal descriptor	TRMDSCR12
authcode	Authorization code	12345
responsecode	Two-character response code from the bank	00
time	Request date	2011-05-31 12:13:15