# Google Pay™

, Google Pay Google. Google Pay https://pay.google.com/about/terms/

## IntellectMoney (Hosted Checkout)

IntellectMoney (, , https://merchant.intellectmoney.ru), Google Pay . : . .

## Google Pay™

, Google Pay™ , IntellectMoney (Hosted Checkout), Google Pay API Google Pay API

- "Google Pay" , Google Pay
- HTTPS TLS 1.2

## Merchant ID Google

, . "Gateway" "Tokenization Method". "Payment Processor or Gateway" "intellectmoney". "Integration Platform Type" "Web" "Android" .

. IntellectMoney Google Pay API "Pay". Google https://developers.google.com/pay/api/web, . Google https://developers.google.com/pay/api/web/guides/brand-guidelines - https://developers.google.com/pay/api/web/guides/test-and-deploy/integration-checklist.

. Ajax (/GooglePay/Payment) , .

```
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"><
/script>
<script src="/js/googlepay.js"></script>
<script async src="https://pay.google.com/gp/p/js/pay.js" onload="onGooglePayLoaded()"></script>
<script>
        var googlePayAmount = "10.00";
        var googlePayCurrency = "RUB";
</script>
<div id="google-pay-button" style="padding: 50px;" />
```

:

- gateway: "intellectmoney"
- merchantId: , Google
- gatewayMerchantId: , IntellectMoney
- allowedCardNetworks = ["AMEX", "DISCOVER", "INTERAC", "JCB", "MASTERCARD", "VISA"]
- allowedCardAuthMethods = ["PAN_ONLY", "CRYPTOGRAM_3DS"];

**googlepay.js**

```
const baseRequest = {
    apiVersion: 2,
    apiVersionMinor: 0
};

const allowedCardNetworks = ["AMEX", "DISCOVER", "INTERAC", "JCB", "MASTERCARD", "VISA"];
const allowedCardAuthMethods = ["PAN_ONLY", "CRYPTOGRAM_3DS"];
const cardPaymentMethodType = 'CARD';
let paymentsClient = null;

function getGoogleIsReadyToPayRequest() {
    return Object.assign(
        {},
        baseRequest,
        {
```

```
            allowedPaymentMethods: [
                {
                    type: cardPaymentMethodType,
                    parameters: {
                        allowedAuthMethods: allowedCardAuthMethods,
                        allowedCardNetworks: allowedCardNetworks
                    }
                }
            ]
        }
    );
}

function getGooglePaymentsClient() {
    if (paymentsClient === null) {
        paymentsClient = new google.payments.api.PaymentsClient({ environment: 'PRODUCTION' });
    }
    return paymentsClient;
}

function onGooglePayLoaded() {
    const paymentsClient = getGooglePaymentsClient();
    paymentsClient.isReadyToPay(getGoogleIsReadyToPayRequest())
        .then(function (response) {
            if (response.result) {
                addGooglePayButton();
            }
        })
        .catch(function (err) {
            // show error in developer console for debugging
            console.error(err);
        });
}

function addGooglePayButton() {
    const paymentsClient = getGooglePaymentsClient();
    const button = paymentsClient.createButton({ onClick: onGooglePaymentButtonClicked });
    document.getElementById('google-pay-button').appendChild(button);
}

function onGooglePaymentButtonClicked() {
    const paymentDataRequest = Object.assign(
        {},
        baseRequest,
        {
            merchantInfo: {
                merchantId: '12345678901234567890' //,   Google
            },
            allowedPaymentMethods: [
                {
                    type: cardPaymentMethodType,
                    parameters: {
                        allowedAuthMethods: allowedCardAuthMethods,
                        allowedCardNetworks: allowedCardNetworks
                    },
                    tokenizationSpecification: {
                        type: "PAYMENT_GATEWAY",
                        parameters: {
                            gateway: "intellectmoney",
                            gatewayMerchantId: "432143" // ,   IntellectMoney
                        }
                    }
                }
            ],
            transactionInfo: {
                totalPriceStatus: 'FINAL',
                totalPrice: googlePayAmount,
                countryCode: 'RU',
                currencyCode: googlePayCurrency
            }
        }
```

```
    );

    const paymentsClient = getGooglePaymentsClient();
    paymentsClient.loadPaymentData(paymentDataRequest)
        .then(function (paymentData) {
            processPayment(paymentData);
        })
        .catch(function (err) {
            // show error in developer console for debugging
            console.error(err);
        });
}

function processPayment(paymentData) {
    console.log(paymentData);
    var status;
    $.ajax({
        cache: false,
        dataType: "json",
        type: "post",
        url: googlePayApiProcessUrl + "/GooglePay/Payment",
        data: "paymentTokenDataJson=" + encodeURIComponent(JSON.stringify(paymentData.paymentMethodData.
tokenizationData.token)),
        complete: function (data) {
            status = 'complete';
            if (data.Result.OperationId) {
                // check payment processing state via getBankCardPaymentState
            }
        },
        error: function (e) {
            status = 'error';
        },
        success: function (e) {
            status = 'success';
        }
    });
}
```

, " G Pay"

payment IntellectMoney ( createInvoice()), .

**API**

API SSL-, . API

POST- URL:

https://api.intellectmoney.ru/merchant/latest/googlepay/pay

:

| | | | |
|---|---|---|---|
| | invoiceId | | - IntellectMoney, . “ ” |
| | returnUrl | | , 3DS- |
| IP | ipAddress | | IP |
| | paymentTokenDataJson | | , Google Pay API |

```
Post /merchant/latest/googlepay/pay
HTTP/1.1 Host: api.intellectmoney.ru
Content-Type: application/x-www-form-urlencoded; charset=utf-8

invoiceId = 3496318551
returnUrl = https://mysite.com/returnUrl
ipAddress = 10.10.10.10
paymentTokenDataJson = {"signature":"MEUCIZ29vZ2xlIHBheSBkZWNvZGVkIHNpZ25hdHVyZSBkYXRhIChiaW5hcn...", ... }
```

```
<Response xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <OperationState>
                <Code>0</Code>
                <Desc> </Desc>
        </OperationState>
        <OperationId>b294b231-9a77-427a-9866-f111e4c88515</OperationId>
        <EshopId>0</EshopId>
        <Result>
                <State>
                        <Code>0</Code>
                        <Desc> .</Desc>
                </State>
                <OperationId> </OperationId>
        </Result>
</Response>
```

, .

IntellectMoney getBankCardPaymentState()

Android, PaymentData IntellectMoney Google Pay API "Pay". Google https://developers.google.com/pay/api/android/, . Google https://developers.google.com/pay/api/android/guides/brand-guidelines - https://developers.google.com/pay/api/android/guides/test-and-deploy/integration-checklist.

Google Pay API Google https://github.com/google-pay/android-quickstart

:

- gateway: "intellectmoney"
- gatewayMerchantId: , IntellectMoney

**Kotlin**

```
private fun gatewayTokenizationSpecification(): JSONObject {
    return JSONObject().apply {
        put("type", "PAYMENT_GATEWAY")
        put("parameters", JSONObject(mapOf(
                "intellectmoney",
                "eshopId")))
    }
}
```

- allowedCardNetworks = ["AMEX", "DISCOVER", "INTERAC", "JCB", "MASTERCARD", "VISA"]
- allowedCardAuthMethods = ["PAN_ONLY", "CRYPTOGRAM_3DS"];

**Kotlin**

```
private val allowedCardNetworks = JSONArray(listOf(
        "AMEX",
        "DISCOVER",
        "INTERAC",
        "JCB",
        "MASTERCARD",
        "VISA"))

private val allowedCardAuthMethods = JSONArray(listOf(
        "PAN_ONLY",
        "CRYPTOGRAM_3DS"))
```

paymentData.getPaymentMethodToken().getToken() POST- URL:

https://api.intellectmoney.ru/merchant/latest/googlepay/pay

:

| | | | |
|---|---|---|---|
| | invoiceId | | - IntellectMoney, . “ ” |
| | returnUrl | | , 3DS- |
| IP | ipAddress | | IP |
| | paymentTokenDataJson | | , Google Pay API |

```
Post /merchant/latest/googlepay/pay
HTTP/1.1 Host: api.intellectmoney.ru
Content-Type: application/x-www-form-urlencoded; charset=utf-8

invoiceId = 3496318551
returnUrl = https://mysite.com/returnUrl
ipAddress = 10.10.10.10
paymentTokenDataJson = {"signature":"MEUCIZ29vZ2xlIHBheSBkZWNvZGVkIHNpZ25hdHVyZSBkYXRhIChiaW5hcn...", ... }
```

```
<Response xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <OperationState>
                <Code>0</Code>
                <Desc> </Desc>
        </OperationState>
        <OperationId>b294b231-9a77-427a-9866-f111e4c88515</OperationId>
        <EshopId>0</EshopId>
        <Result>
                <State>
                        <Code>0</Code>
                        <Desc> .</Desc>
                </State>
                <OperationId> </OperationId>
        </Result>
</Response>
```

, .

IntellectMoney getBankCardPaymentState()

**API**

API SSL-, . API

## 3DS-

( PAN_ONLY) 3ds-. :

- getBankCardPaymentState() PaymentStep=3, ( Form3DS)
- 3ds- PaymentStep=0, , resultUrl

- IntellectMoney (Hosted Checkout)
- Google Pay™
- 
- Merchant ID Google
- 
  - 
  - 
- 
- 3DS-