Google Pay™

Documentation

Change Log

Version	Date	Description	Pages	Author
1.0	10/3/2022	First Version	1 - 10	Bar Bashari
1.1	16/03/2022	Adding XML API Adding Limitations	3-5 , 8-12	Tzvika Velich

Introduction

Hyp lets merchants take advantage of fast, cashless payments for products and services via a mobile device and web browsers with Google Pay™.
Google Pay™ is a fast, simple way to pay at millions of places - on sites, in apps, and in stores. It brings together everything you need at checkout and protects your payment info with multiple layers of security.

Types of integrations

Hyp supports 3 types of integration methods for merchants with Google Pay™

1. **WEB API:**
   Merchants that wish to integrate themselves with the Google Pay™ WEB API.
   This type of integration is suitable for merchants that:
   - Wish to have better control how/where & when Google Pay™ button will be used in the application.
   - Work with more than one PSP (payment service provider).

   On the other hand, for this option the merchant is required to complete the integration with Google Pay™ WEB API and request Google for production access.
2. **Android API:**
   Merchants that wish to integrate themselves with the Google Pay™ Android API.
   This type of integration is suitable for merchants that:
   - Wish to use the native mobile UI/UX for their App users.
   - Wish to have better control how/where & when Google Pay™ button will be used in the application.
   - Work with more than one PSP (payment service provider).

   On the other hand, for this option the merchant is required to complete the integration with Google Pay™ Android API and request Google for production access.
3. **Hyp Payment page:**
   Merchants that don't require the above and just need that the Google Pay™ button will displayed in Hyp Payment page.
   - No additional integration is required by the merchant.
   - After quick setup at Hyp Google Pay™ button will be enable in Hyp Payment page.

The Goal

This document is intended for merchants that wish to **integrate themselves with the Google Pay™ API** (as explained above for integration types 1,2).
The document details the technical and operational assessments required of a merchant to charge transactions via **Hyp** gateway API with the Google Pay™ button (in web or android applications).
The document assumes that the merchant has completed integration to the Google Pay™ API.
This document is intended for the software, the product, and the IT people in the organization.

How it works

After full integration with Google Pay™, a payment button via Google Pay™ will appear in your checkout page, then the payment flow is as follows:

1. The user clicks the Google Pay™ payment button and sees a payment sheet with a list of supported payment methods.
2. The user selects a payment method and Google Pay™ securely returns a payment token for that method to your website.
3. Your website submits the payment token, along with details about the purchase, to its backend.
4. To execute the payment, the backend processes the purchase and sends the payment token to **Hyp** via the Hyp's doDeal API.

Compatibility & Limitations

1. **Hyp** accepts Google Pay™ only for Israeli merchants and Israeli acquirers.
2. In Israel Google Pay™ should be only used along with Ashrait EMV terminals.
3. For this **type** of integration, the Google Pay™ button will be available only on Android devices.
   note: for that matter you'll be asked in the integration section below to set the allowed auth methods to accept only "CRYPTOGRAM_3DS".
4. We support the following card networks:
   (please see in the integration section below how to set up these card networks)
   - Mastercard
   - Visa
   - American Express
5. One phase/charge transaction is supported (or in other words – J4/AutoComm)
6. Two phase/auth & capture transactions are supported (or in other words – J5/Verify & J4/autoComm)
7. Refund & cancel transactions are supported
8. COF transactions are not supported – HYP tokens (cardId) that were created in Google Pay transaction cannot be used for future transactions.

Implementing the integration

Setup

Prerequisites

You need:

1. An active account & terminal at **Hyp** gateway
2. A Google merchant account and merchant Id for production
3. **Hyp** to enable Google Pay™ on your account
4. To make sure that Ashrait terminals are defined as "allowed for Internet transactions".
5. To make sure that the acquirer "Sapak" number is defined as "allowed for Internet transactions"
6. adhere to the Google Pay™ APIs Acceptable Use Policy and accept the terms defined in the Google Pay API Terms of Service.

Hyp Gateway Merchant ID

When Google Pay™ is enabled on your **Hyp** account, use the terminal number as your own **gatewayMerchantID**.

Set up Google Pay™

For detailed instructions on how to configure Google Pay™ for web payments, see here.

For detailed instructions on how to configure Google Pay™ for in-app payments, see here.

There is no integration required for Hosted Payment Page integration, but you just need to make sure you enable the Google Pay™ on your **Hyp** account.

Integrate with Google Pay™

For web integration, perform the following steps to get started with the Google Pay™ API:

1. link out to: https://developers.google.com/pay/api/web/guides/tutorial and implement the Google Pay™ API in your site.
2. Set your **environment** according to TEST/PRODUCTION and use your Google pay™ **Merchant ID** you got from Google Pay Business Console (for production):

```
{
  environment: "TEST",
  merchantInfo: {
    merchantName: "your terminal name",
    merchantId: "your google pay marchant id"
  },
  paymentDataCallbacks: {
    onPaymentDataChanged: onPaymentDataChanged,
    onPaymentAuthorized: onPaymentAuthorized
  }
}
```

3. set your **gateway** as "*hyp*" and **gatewayMerchantId** as your hyp terminal number:

```
"tokenizationSpecification": {
  "type": "PAYMENT_GATEWAY",
  "parameters": {
    "gateway": "hyp",
    "gatewayMerchantId": "your terminal id in Hyp"
  }
}
```

4. Define **Hyp** supported authentication method:

```
const allowedCardAuthMethods = ["CRYPTOGRAM_3DS"];
```

5. Define **Hyp** supported payment card networks:

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

For in-app integration, perform the following steps to get started with the Google Pay™ API:

1. link out to : https://developers.google.com/pay/api/android/guides/tutorial and implement the Google Pay™ API in your app.
2. Set your **environment** according to TEST/PRODUCTION and use your Google Pay™ **Merchant ID** you got from Google Pay Business Console (for production).
3. set your **gateway** as "*hyp*" and **gatewayMerchantId** as your hyp terminal number:

```
private static JSONObject getGatewayTokenizationSpecification() throws JSONException {
  return new JSONObject() {{
    put("type", "PAYMENT_GATEWAY");
    put("parameters", new JSONObject() {{
      put("gateway", "hyp");
      put("gatewayMerchantId", "your terminal id in hyp");
    }});
  }};
}
```

4. Define **Hyp** supported authentication method:

```
private static JSONArray getAllowedCardAuthMethods() {
  return new JSONArray()
      .put("CRYPTOGRAM_3DS");
}
```

5. Define **Hyp** supported payment card networks:

```
private static JSONArray getAllowedCardNetworks() {
  return new JSONArray()
      .put("AMEX")
      .put("DISCOVER")
      .put("MASTERCARD")
      .put("VISA");
}
```

Integrate with Hyp XML API for payment (the doDeal command)

All the usual parameters that sent in the doDeal request, will be sent here as well without change, except the of following:

1. Add the additional XML Tags **walletData & walletToken**.
2. The cardNo XML Tag will be populated with a constant value "GOOGLEPAY"
3. The cardExpiration Tag should be omitted

```
<ashrait>
  <request>
    <command>doDeal</command>
    <language>Eng</language>
    <version>2000</version>
    <doDeal>
      <validation>{{transaction jcode value here, e.g: Verify}}</validation>
      <creditType>RegularCredit</creditType>
      <transactionType>Debit</transactionType>
      <total>{{total value here, e.g: 15000}}</total>
      <terminalNumber>{{terminal number value here, e.g: 0882817010}}</terminalNumber>
      <transactionCode>Internet</transactionCode>
      <cardNo>GOOGLEPAY</cardNo>
      <currency>{{transaction currency here, e.g: ILS}}</currency>
      <walletData>
                <walletToken>{"signature":"MEUCIQDhJ+oEJPWehLIsfbgqZpU/kfl76uB8Hej9ZNqIPZPOPAIgY4LVxUbMXK+Zwg
                m6lNjPKi0glmsFKLCIStFuZCi6E0I\u003d","intermediateSigningKey":{"signedKey":"{\"keyValue\":\"MFkwEwYHK
                oZIzj0CAQYIKoZIzj0DAQcDQgAE09EkTgH+wdZFuUxMstOhaU2/aW/jWeJ/CSlUDIru4VALhfoQ/NlGsQ2HmX1Gk0
                WQyMsfgXXSCdmjlzLOXPn+OA\\u003d\\u003d\",\"keyExpiration\":\"1642788088880\"}","signatures":["MEYC
                IQDYOHBu5+/OJXN0IoM79meNTZy0ktfqqhrjHZDpD/ky6wIhAJWIRofLkJDh/WuU4Y0C2JoFiC5ft0OL+hAfjOyL5HR
                e"]},"protocolVersion":"ECv2","signedMessage":"{\"encryptedMessage\":\"Hsor2M32v7Im4OG184Fw92FkNZ//
                vOfII7Dc6Mv3t/CsZ3nSvAsiXgoYOJimoq+tCUzMjDfcIM/G0JlPK93W/thPDGt0NL2gXXYw9YFHwrbb3+8GKk0uv8q
                zcx0P6AZDPwAP500PoweDpTtzHNnzRsuI8l5VpQX3pNgezMIH03kcZdNOTQbuWG6iTsOeY52zCRP8EQglJId4IrLIe
                Ch4z+8fFBlpOql3xYKuo64hrjst4gCWwqDyzc48pykr1XE1yXewsB2olp5AoLUI9zKRr9PX6E4uQ1KmLGSagv51RPuJD
                OnQLlp9JL880Dtt48D9M2b40ln3kCPoWUjLwyb3rhkiKE2kurYBB7y4xdQZ33HLA16NUuxXXzlq+ta3w/hh0a5aZQ0
                G5oUc0SQynbKWAmwj0zuP9iVRabdDAeWGC6fg7hmeaDcU8fvhWoClKtjy7EULcI7JIy0rfXzc2OgTfgN87Z0mCXCIk
                SeeSJPxVd617MbxNbfcI6PjlVWKa2Wp/DLb0EMILW1Hr7eqxzvOGrl8H8U0M5ApRrAtIR0NtWLfx/6ajMkRj+HAsZF
                oAjp7gdkK2SjXC1wI9WCbPrsCZmHQQiZA2Q3Pu4Mmmujq6hI3uIp0ETzm+9mzFpnDzol5Nz3yfYm86EJB0Q\\u00
                3d\\u003d\",\"ephemeralPublicKey\":\"BMjb6m+MIAxs/vNKiVW11Ly6tWoEtY/HW7joDzsbG8kzQP1pLEWID7F
                SEOt6SGNNs5m8gv6FeWgiSPRTREFGRQU\\u003d\",\"tag\":\"3ZpexlhtGG1ILR0WNBK510ycKbBiAkXgHFQ8nK5
                3FtI\\u003d\"}"}
        </walletToken>
      </walletData>
    </doDeal>
  </request>
</ashrait>
```

The response need to look like this:

```
<ashrait>
    <response>
        <command>doDeal</command>
        <dateTime>2022-03-17 08:37</dateTime>
        <requestId />
        <tranId>73977663</tranId>
        <result>000</result>
        <message>Permitted transaction</message>
        <userMessage>Permitted transaction</userMessage>
        <additionalInfo>Host Result Remote 00-SUCCESS </additionalInfo>
        <version>2000</version>
        <language>Eng</language>
        <doDeal>
            <status>000</status>
            <statusText>Permitted transaction</statusText>
            <extendedStatus />
            <extendedStatusText />
            <extendedUserMessage />
            <terminalNumber>0882818012</terminalNumber>
            <cardId>1065552051077840</cardId>
            <cardBin>520424</cardBin>
            <cardMask>520424******7840</cardMask>
            <cardLength>16</cardLength>
            <cardNo>xxxxxxxxxxxx7840</cardNo>
            <cardName />
            <cardExpiration>1227</cardExpiration>
            <cardType code="99">Foreign</cardType>
            <extendedCardType code="0">Credit</extendedCardType>
            <lifeStyle />
            <creditCompany code="0">Foreign</creditCompany>
            <cardBrand code="1">Mastercard</cardBrand>
            <cardAcquirer code="1">Isracard</cardAcquirer>
            <serviceCode />
            <transactionType code="01">RegularDebit</transactionType>
            <creditType code="1">RegularCredit</creditType>
            <currency code="1">ILS</currency>
            <baseCurrency />
            <baseAmount />
            <transactionCode code="52">Internet</transactionCode>
            <total>1000</total>
            <firstPayment />
            <periodicalPayment />
            <numberOfPayments />
            <clubId />
```

```
            <validation code="5">Verify</validation>
            <idStatus code="0">Absent</idStatus>
            <cvvStatus code="3">NotValidated</cvvStatus>
            <authSource code="2">CreditCompany</authSource>
            <authNumber>3805468</authNumber>
            <fileNumber />
            <slaveTerminalNumber />
            <slaveTerminalSequence />
            <eci>5</eci>
            <clientIp />
            <email />
            <cavv code="" />
            <user />
            <addonData />
            <supplierNumber>2441070</supplierNumber>
            <id />
            <shiftId1 />
            <shiftId2 />
            <shiftId3 />
            <shiftTxnDate />
            <cgUid>73977663</cgUid>
            <cardHash />
            <cg3dsData>
                <cg3dsVersion>2</cg3dsVersion>
            </cg3dsData>
            <acquirerData>
                <walletSupplierId>2</walletSupplierId>
                <gateway>AshraitEmv</gateway>
                <acquirerTranType>01</acquirerTranType>
                <mcc>4121</mcc>
                <acquirerResponseId>693301800034</acquirerResponseId>
                <avsResponse code="0">Absent</avsResponse>
                <acquirerTranCode>52</acquirerTranCode>
            </acquirerData>
            <ashraitEmvData>
                <uid>22031708373008828186631</uid>
                <authCodeCreditCompany
code="1">CreditCompanyAuthorized</authCodeCreditCompany>
                <idFlag>0</idFlag>
                <manufId>CGD</manufId>
                <catLevel>0</catLevel>
                <manufUse>001101</manufUse>
                <ashVersion>x</ashVersion>
                <ashTermType>0</ashTermType>
                <emvResponseCode>00</emvResponseCode>
                <deviceStatus>1111000000</deviceStatus>
                <ashReasonText>SUG_ISKA, BAKASHA_LEISHUR_LELO_ISKA</ashReasonText>
```

```
                <authCodeAcquirer code="0">NoAuthNumber</authCodeAcquirer>
                <isDoReverseDeal>0</isDoReverseDeal>
            </ashraitEmvData>
            <walletData>
                <walletType>googlePay</walletType>
                <cardHolderAuthenticated>true</cardHolderAuthenticated>
            </walletData>
        </doDeal>
    </response>
</ashrait>
```

Relevant error codes

Code	Reason	Message text
388		Transaction parameters do not match approved parameters
389		
390		
401		

Merchant End to End Testing

Testing with a tokenized card (Google Pay™ encrypted JSON token contains a token)

1. Add a 'contactless' enabled credit card using Google Pay™ app in your android phone.
2. Follow the Google Pay™ instructions to create a test app to generate a test cryptogram.
    - For in-app integration, please refer to https://developers.google.com/pay/api/android/guides/setup & https://developers.google.com/pay/api/android/guides/tutorial.
    - For web integration, please refer to https://developers.google.com/pay/api/web/guides/setup & https://developers.google.com/pay/api/web/guides/tutorial.
3. Modify to use "***hyp***" as the gateway and populate the "hyp terminal number" as gatewayMerchantId in GATEWAY_TOKENIZATION_PARAMETERS.
4. Google Pay™ returns a payment token response in JSON format:

{"signature":"MEUCIQDhJ+oEJPWehLIsfbgqZpU/kfl76uB8Hej9ZNqIPZPOPAIgY4LVxUbMXK+Zwgm6lNjPKi0glmsFKLCIStFuZCi6E0I\u003d","intermediateSigningKey":{"signedKey":"{\"keyValue\":\"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE09EkTgH+wdZFuUxMstOhaU2/aW/jWeJ/CSlUDIru4VALhfoQ/NlGsQ2HmX1Gk0WQyMsfgXXSCdmjlzLOXPn+OA\\u003d\\u003d\",\"keyExpiration\":\"1642788088880\"}","signatures":["MEYCIQDYOHBu5+/OJXN0IoM79meNTZy0ktfqqhrjHZDpD/ky6wIhAJWIRofLkJDh/WuU4Y0C2JoFiC5ft0OL+hAfjOyL5HRe"]},"protocolVersion":"ECv2","signedMessage":"{\"encryptedMessage\":\"Hsor2M32v7Im4OG184Fw92FkNZ//vOfII7Dc6Mv3t/CsZ3nSvAsiXgoYOJimoq+tCUzMjDfcIM/G0JlPK93W/thPDGt0NL2gXXYw9YFHwrbb3+8GKk0uv8qzcx0P6AZDPwAP500PoweDpTtzHNnzRsuI8l5VpQX3pNgezMIH03kcZdNOTQbuWG6iTsOeY52zCRP8EQglJId4IrLIeCh4z+8fFBlpOql3xYKuo64hrjst4gCWwqDyzc48pykr1XE1yXewsB2olp5AoLUI9zKRr9PX6E4uQ1KmLGSagv51RPuJDOnQLlp9JL880Dtt48D9M2b40ln3kCPoWUjLwyb3rhkiKE2kurYBB7y4xdQZ33HLA16NUuxXXzlq+ta3w/hh0a5aZQ0G5oUc0SQynbKWAmwj0zuP9iVRabdDAeWGC6fg7hmeaDcU8fvhWoClKtjy7EULcI7JIy0rfXzc2OgTfgN87Z0mCXCIkSeeSJPxVd617MbxNbfcI6PjlVWKa2Wp/DLb0EMILW1Hr7eqxzvOGrl8H8U0M5ApRrAtIR0NtWLfx/6ajMkRj+HAsZFoAjp7gdkK2SjXC1wI9WCbPrsCZmHQQiZA2Q3Pu4Mmmujq6hI3uIp0ETzm+9mzFpnDzol5Nz3yfYm86EJB0Q\\u003d\\u003d\",\"ephemeralPublicKey\":\"BMjb6m+MIAxs/vNKiVW11Ly6tWoEtY/HW7joDzsbG8kzQP1pLEWID7FSEOt6SGNNs5m8gv6FeWgiSPRTREFGRQU\\u003d\",\"tag\":\"3ZpexlhtGG1ILR0WNBK510ycKbBiAkXgHFQ8nK53FtI\\u003d\"}"}

5. Create a XML API doDeal payment request. In this request you must pass the Google Pay™ payment token response value in the w*alletToken* field as received from Google Pay™ API.
6. Send this request to **Hyp** for processing.

Hyp support

To contact us about API & integration Issues Use these emails:
Cg-support@hyp.co.il
or call: *6488

See also:

- Google Pay Android integration checklist
- Google Pay Android brand guidelines
- Google Pay Web integration checklist
- Google Pay Web Brand Guidelines.