Google Pay™ Setup

=

Comments
Please send your comments, questions, or suggestions about this document to:

Paragon Payment Solutions
ATTN: Paragon Product Management
2141 E. Broadway Rd, Suite 202,
Tempe, AZ 85282
(800) 884-5208
product@paragonsolutions.com

Contents

Introduction

As part of being a Google Pay supported processor, the following flow is to be used to integrate Google Pay into an app or website. Currently we support One – time token use for secure transactions. Google Pay is also supported in both the REST and SOAP Versions of our API, although using the REST version is encouraged. Supported Authorization methods include 3D – Secure(CRYPTOGRAM_3DS) and or pan and expiration date(PAN_ONLY).

Supported Authorization Methods

Current Supported authorization methods in the U.S
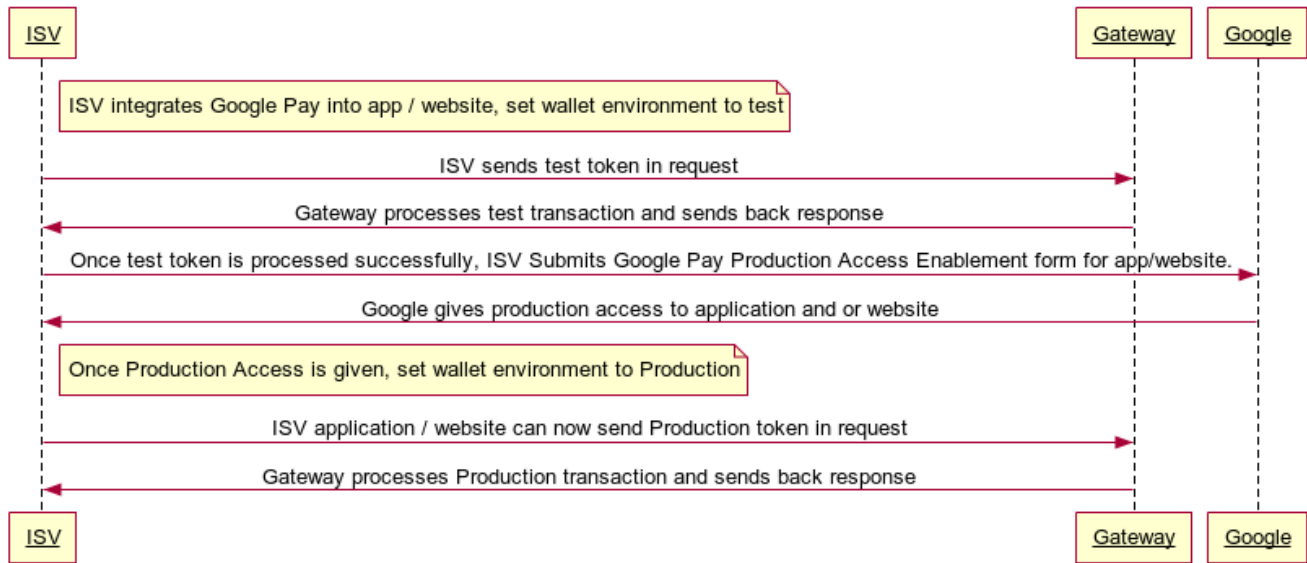- PAN_ONLY
- CRYPTOGRAM_3DS

Supported Card Brands

Current Supported Card Brands in the U.S

- Carte Blanc
- NOVUS
- Discover Card
- American Express
- JCB
- JAL
- Debit
- Visa
- Diners Club
- MasterCard

Sequence Diagram

Google Pay Sequence Diagram

ISV | Gateway | Google

ISV integrates Google Pay into app / website, set wallet environment to test

ISV sends test token in request

Gateway processes test transaction and sends back response

Once test token is processed successfully, ISV Submits Google Pay Production Access Enablement form for app/website.

Google gives production access to application and or website

Once Production Access is given, set wallet environment to Production

ISV application / website can now send Production token in request

Gateway processes Production transaction and sends back response

ISV Setup

ISVs are required to integrate directly to Google Pay from either a mobile application or website. Tutorials and examples of how to integrate Google Pay can be found here.

Branding Requirements:
/   Assets for Android
/   Assets for Website

The ISV will need to pass in the following 2 parameters into the *getTokenization* method:

/   .put("gateway", "paragonpaymentsolutions")
/   .put("gatewayMerchantId", "abcMerchant"));

To learn more about the *getTokenization* method, please go to step 2 in the Tutorials section for either android or web. How to retrieve the merchant identifier can be found here.

Once the integration has been completed, a payment token can be created with Environment set to test. The ISV will need to fill out a production request form, this form can be found at https://services.google.com/fb/forms/googlepayAPIenable/. If integrating a website, the URL must be submitted in the form, if an app is being developed, the Application will need to be in the Google Play Console. The next sections, Sending Request and For Apps includes more information on how to do this.

Sending Request Through Paragon REST API

ISVs need to pass in the Google payment token as a base64 UTF-8 string into the credit card request object. The request will also require at a minimum the following parameters:

- ⁄ Amount
- ⁄ Transaction type
- ⁄ Force duplicate

Billing information is not required. The request should be sent to the endpoint provided by ISV Support.

JSON Request

```
{
  "credit_card": {
    "transaction_type": "sale",
    "force": "true",
    "amount": "1.00",
    "google_pay_payment_data": "eu************************************************="
  }
}
```

Sending Request Through Paragon Mobile SDK

ISVs need to pass in the Google payment token as is into the CreditCardRequest Object that is provided with the Mobile SDK. The request will also require at a minimum the following parameters:

- ⁄ Amount
- ⁄ transaction type
- ⁄ Force duplicate
- ⁄ Username
- ⁄ Password
- ⁄ Gateway ID

Billing information is not required. The endpoint to be used will be provided by ISV Support.

MobileSDK Request

```
paymentMethodData = new JSONObject(json) .getJSONObject("paymentMethodData");
token = paymentMethodData.getJSONObject("tokenizationData").getString("token");

CreditCardRequest reqs = new CreditCardRequest(gatewayEndpoint);
reqs.setValue(CreditCardRequest.UserName, userName);
reqs.setValue(CreditCardRequest.Password, password);
reqs.setValue(CreditCardRequest.GatewayID, "2***0");
reqs.setValue(CreditCardRequest.TransType, "Sale");
reqs.setValue(CreditCardRequest.Force, "true");
reqs.setValue(CreditCardRequest.Amount, "1.00");
reqs.setValue(CreditCardRequest.GooglePay, token);
```

Integration for Websites

To integrate Google Pay into any website, the following should be used as reference for integration:

- / [Google Pay Web Brand Guidelines](#)
- / [Google Pay Web integration checklist](#)

Integration for Apps

For any mobile application being developed for Production, you will need to create a Developer account.

Google Developer Console can be found at: https://developer.android.com/distribute/console/.

Once signed in, select All Applications, Create Application. You will be redirected to the following screen.
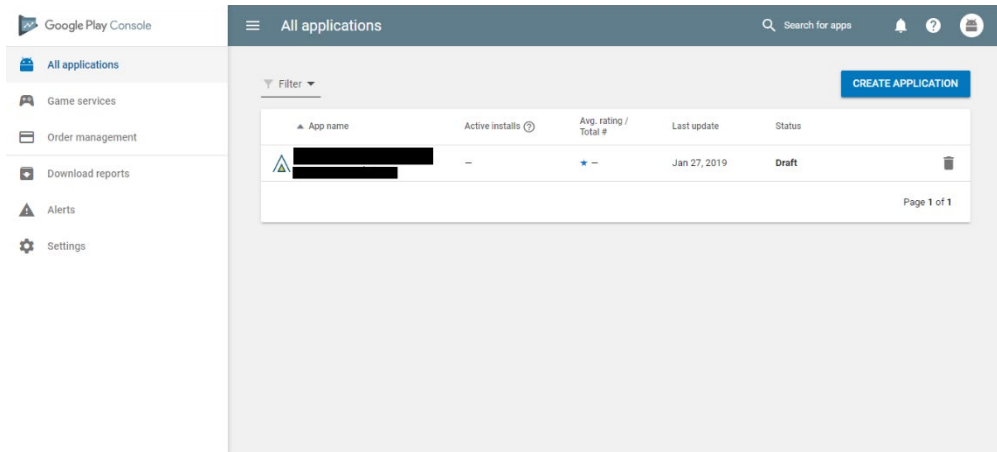
Select the project. You will be redirected to the screen below. In here add the release .apk file in app releases. A Google representative will give you information on what is required after application has been successfully added to the Google Play Console.
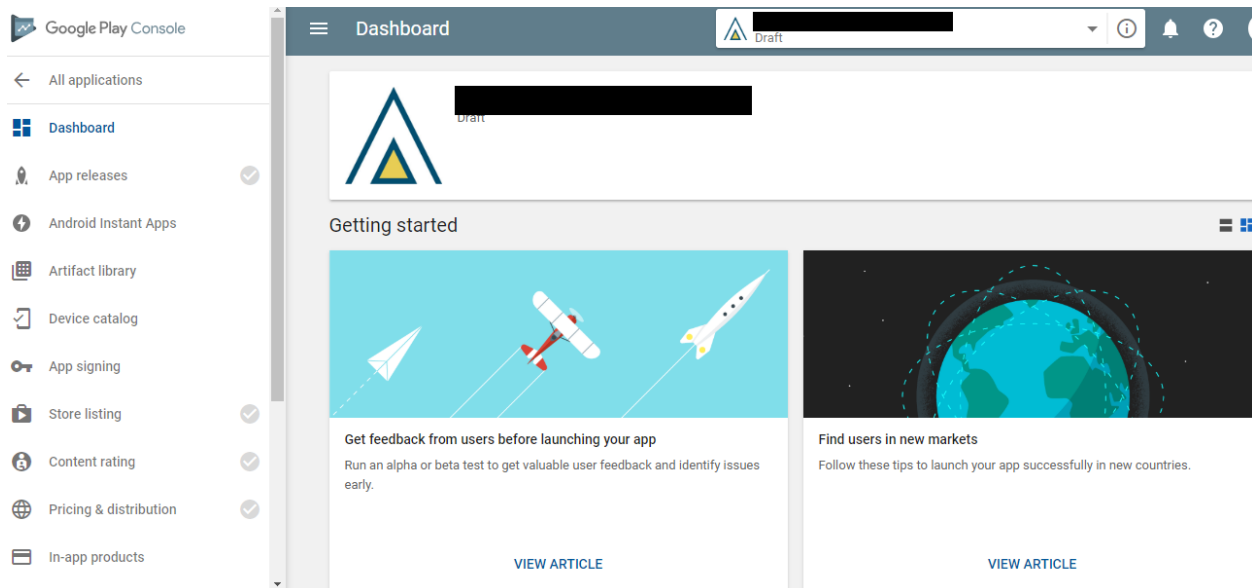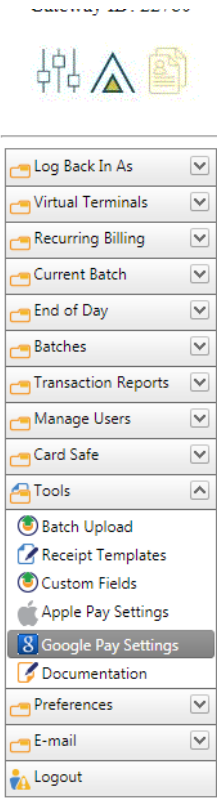
Retrieving Merchant Identifier

To retrieve the merchant identifier, in the merchant or reseller portal, go into Tools -> Google Pay Settings -> Download Merchant Identifier.