

Push Provisioning Step by Step

Android - Push Provisioning to Google Pay

1. Request Access

Before enabling Push Provisioning in the Android application, you'll need to be whitelisted by Google. To initiate this process:

- Visit the **Google Pay Push Provisioning** launch page:
<https://developers.google.com/pay/issuers/apis/push-provisioning/android/launch-process>
- Click "**Request access**" and complete the provided form.

2. Provide Application Details (Upon Approval)

Once your access request is approved, Google will contact you to gather additional information about your application:

- **Screenshots:** Submit screenshots that showcase the placement and functionality of the "Google Pay" button within your app.
- **User Flow Recording:** Provide a screen recording demonstrating the user experience for adding a card to Google Pay.

Important Note: Google Pay integration does not require any additional paid certifications.

3. SDK Configuration

- **Verestro Application Ownership:** If Verestro develops the application, our development team will handle the internal configuration of 3rd party Google Pay SDK.
- **Customer Application Ownership:** If the customer owns the application, customer's development team needs to get the Google Pay SDK and follow official documentation.

iOS - Push Provisioning to Apple Pay

1. Contact Apple

To begin enabling Push Provisioning for **Apple Pay**, initiate contact with Apple directly. Use the following email addresses:

- apple-pay-provisioning@apple.com

- applepayentitlements@apple.com

2. Application Information and Testing

Provide Apple with the details of your App Store application. Verestro will conduct integration testing using the same application (including the same ADAM ID) within the TestFlight platform.

3. FIME Certification (Optional)

While not mandatory, FIME can provide an optional certification process for Apple Pay Push Provisioning. Contact FIME directly for current pricing and instructions. This cost was approximately €3125 at the time of writing.

4. SDK Configuration

- **Verestro Application Ownership:** If Verestro develops the application, our development team will handle the internal configuration of the 3rd party Passkit SDK.
- **Customer Application Ownership:** If the customer owns the application, customer's development team needs to get the SDK of their choice (Apple Pay, Passkit, etc.) and follow their documentation.

Additional Notes

These instructions provide a high-level overview.

For detailed technical guidance and code implementation, please refer to the official Google Pay and Apple Pay developer documentation.

It will be necessary to implement the 3rd party **SDKs** and **APIs** provided by [Verestro](#) and used to encrypt card data and support additional processes in xPay's.

Revision #1

Created 13 February 2026 09:08:10

Updated 13 February 2026 09:08:10