Android SDK Integration Document

Table of Contents

1. Ezetap SDK Introduction

Ezetap is a digital payments platform that makes your payment acceptance simple and configurable. This Universal Payment Acceptance platform is a smart, simple end-to-end payments platform which is unique in its vision and architecture.

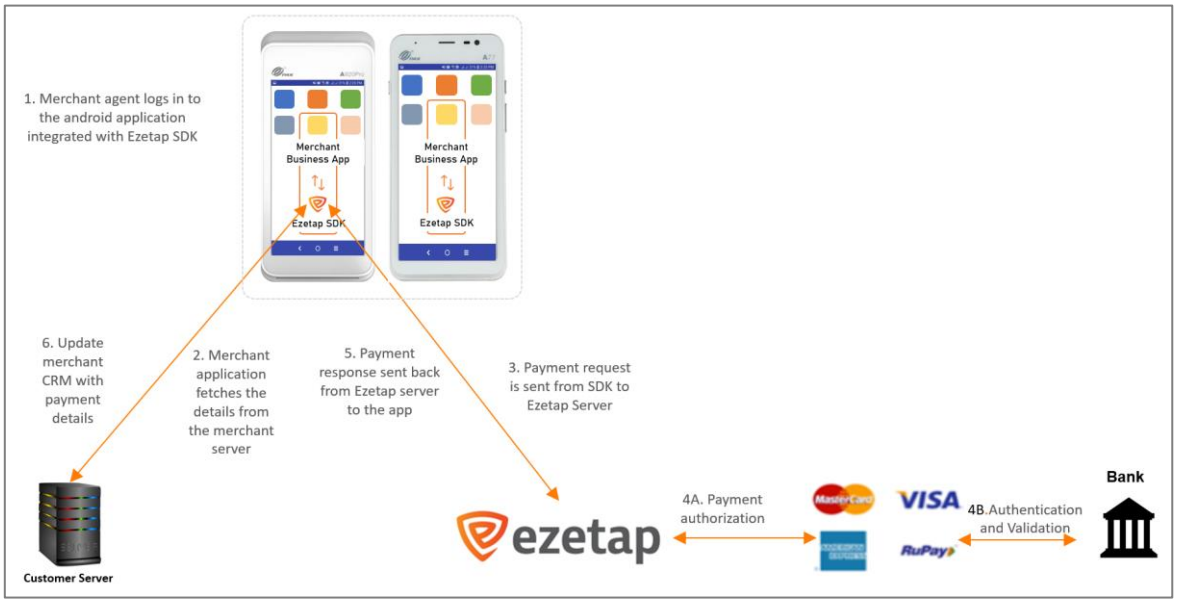
It can be deeply integrated with any enterprise system/sales channel to offer a unified and seamless payments experience to the end customer.

2. Why Ezetap SDK?

- Enterprises who have sales and collection application already built and would like to accept digital Payments via the application. Ezetap's android SDKs are available to be plugged in to the existing application for payments acceptance.
- This integration is fast and easy, and typically takes around 2-3 days of time to complete.
- Enterprises can maintain one single application for business process as well as payments and can future proof their payments capabilities with easy enablement of newer modes of payments.
- NOTE: Ezetap android SDK will only support Android version 5.0 and above. This is due to PCI DSS compliance.

3. What does Ezetap Android SDK do?

Ezetap SDK enables enterprises field application for universal payments acceptance. The SDK helps you to integrate with the Service Application of Ezetap by calling an API (Which can be initiated via a single **"Pay"** button on your application), Ezetap Service Application manages all interactions between the Card Device, Ezetap server and encapsulates a smooth user experience during the entire payment cycle. The below figure showcases the flow of payment using Ezetap SDK.



4. Getting started with Android SDK Integration

We recommend that you get started with integration in demo environment, validate the flow, and use cases on your application and then move to the Production environment.

You will receive the below details from Ezetap Solution Consultant for integration with demo environment.

- Appkey for Authentication
- Credentials to access the demo portal

If you haven't received these details yet, please share the below details to respective Sales or Solution Consultant for an account to be created in Demo Environment.

- Your Organisation's Name
- Your Organisation's Address
- Contact Number & Email ID

Demo device (in case of card payments)- should be available with you from banks. The device should be in debug mode to do integration and testing in demo.

(Devices which are in debug mode have a watermark - "*DEBUG only, Not for Commercial*", at the bottom right corner of the screen as shown in the image below.

5. How to include Payments SDK in your Android Application

Ezetap payments SDK can be integrated with different types of android SDK such as Native, react native, Cordova, etc. Integrating the SDK will allow you to access Ezetap services in your application.

For step-by-step guide with code and other details, please refer to our GitHub portal here - https://github.com/ezetap/android-payments-sdk. You will find the following details here:

- What you need for Integration
- Sample App
- Documentation to create Cordova plugin for Android

6. How SDK APIs work?

1. Include the SDK in your mobile application to collect digital payments.
2. SDK interfaces with Ezetap Service Application, which has a pet name Service App @Ezetap.
3. The SDK inspects the availability of Service App on Android device and if not present it will be installed at run-time.
4. Service App interfaces with the Card Device (in card of card payments) and Ezetap Servers to finish payment processing and notifies the final status to SDK/Client App.

For more details, please refer to our portal - http://d.eze.cc/ezetapsdk/ezetap-native-android-sdk/getting_started/how/. You will find the following details

- Invoking Ezetap API
- Handling Responses
- Sample code to fetch the response
- Response Structure
- Sample Response

7. Step 1: Initializing the SDK

The first API that needs to be integrated with is the “Initialize” API, this API performs the following 3 key activities and 1 optional activity:

- Initializes the SDK with global configuration settings
- Connects to the appropriate Ezetap server based on Application mode (AppMode) = DEMO/ PROD
- Connects to the appropriate merchant account, this depends on the app key entered
- Optionally invokes prepare Device to initialize the device (card reader) with the updated encryption keys from the corresponding bank

Initialize is the first method to be called. It is recommended to call this method post user logs in to your application or (if login is not available) when he reaches to home screen of your application.

For step-by-step guide to initialize the SDK, please refer to our portal here - http://d.eze.cc/ezetapsdk/ezetap-native-android-sdk/getting_started/initialize_sdk/

You will find the following details:

- How to call initialize API
- How to prepare input for initialize API
- How to invoke the initialize API
- How to handle the response of Initialize API
- Sample Request and Response

8. Universal Pay API for Payment Transaction

Ezetap has a **universal pay API** through which all the payment modes (that is enabled for the merchant) can be invoked through a single API call. With this API there will be no need of calling the individual methods for different payment modes (like Card, Remote Pay, QR etc).

Please refer to our portal here - http://d.eze.cc/ezetapsdk/ezetap-native-android-sdk/initiate_payment/pay_api/.

You will find the following details:

- How to prepare Input for the Pay API
- How to invoke Pay API
- How to handle the response of Pay API
- Basic structure of Pay API response
- Sample Pay API response for payment by Card
- How to handle payment failures

9. Card Payment API for payment via CARD

In addition to the universal Pay API described above, Card Payment API is also available for card-based payment transaction.

Please refer to our portal here - http://d.eze.cc/ezetapsdk/ezetap-native-android-sdk/other_operations/card_api/

You will find the following details:

- How to invoke Card Payment API
- JSON request
- Sample Request
- Handling the payment response
- Sample Response

10. How to Identify a successful Card transaction

For Card transaction, please rely on the Status Field in the Pay API or Card Payment API response to identify a successful transaction

Status	What it means
Authorized	Transaction has been successfully executed
Failed	Transaction has not been executed and has failed; the money won't be
	deducted in this scenario
Voided	The transaction was authorized, and which is now has been voided.
Refunded	The transaction was completed and after which it was refunded

11. APIs for Other Payment Modes such as UPI, QR, etc.

APIs for other payment modes such as UPI, Cash, Cheque, etc. are also available. Please refer below for details of all such APIs and respective links to our portal.

API For Payment Modes	Link to Portal
Cash	Link
Cheque	Link
UPI	Link
Remote Pay	Link
QR Code	Link
Wallet	Link

Upon clicking the link provided in the table above, you will find below information

- How to prepare input for API
- How to invoke the API
- How to handle response of the API
- Sample Request & Response

12. Additional APIs for Payment and other related operations

Some other useful APIs that may be required for payment and related operations are also available. Please refer below for details of all such APIs and respective links to our portal.

API	Usage	Link to Portal
Service Fee	To add Service Fee	Link
Accepting Meal Cards	To accept payment via meal cards	Link
Void Payment API	To process Refund on same day	Link
Check for Incomplete Payment	To check for the incomplete transaction	Link
Fetch Payment Details	To retrieve the details of a payment transaction	Link
Sending Receipts	To send e-receipts via SMS/e-mail	Link
Print Receipts	To print receipt/charge-slip/custom slip	Link
Close SDK	To exit from Ezetap SDK	Link

Upon clicking the link provided in the table above, you will find below information

- How to prepare input for API
- How to invoke the API
- How to handle response of the API
- Sample Request & Response

13. Print custom receipts, bills, invoice, in any format

Ezetap provides custom print SDK which allows printing of custom bills, invoices receipts in addition to the chargeslip. Bills can be printed via the device in any format, layout of our choice.

Text Size & Font:

- Text Size: setTextSize(TypedValue.COMPLEX_UNIT_PX,15sp)
- font family: lato_bold

For sample code, SDK, please refer to the link - https://demo.ezetap.com/ezetapsdk/printer/ezetap-printer-sdk-package.zip

14. Error Codes

Typically, you will notice 4 types of errors during transactions:

- Common Errors
- Server Errors
- PG Errors
- SDK Errors

Full list of these error codes is available on our portal here - http://d.eze.cc/ezetapsdk/ezetap-native-android-sdk/misc/failures/

Please refer to this for prominent error codes and their definitions.

15. Steps for Go Live

Testing the Integration

A detailed testing scenario is crucial and critical step before the roll-out to production, you can test all payment related integration features End-to-End including failure cases, payment gateways errors etc. To facilitate this on our demo environment we do simulation of error mapped to certain amounts. Below you can find the amounts and mapped error cases.

Best Practice: Ezetap recommends testing at least 50% of these scenarios to conclude your UAT

Please refer to our portal here - http://d.eze.cc/ezetapsdk/ezetap-native-android-sdk/misc/testing/

You will find the details of amounts to be passed to simulate a specific error.

Go-Live

Once you closed your integration and testing on our DEMO environment, you should consult the Ezetap team and get your integration verified. Post we have confirmed and tested the integration and various scenarios; you will be eligible to go live. Here are the steps you need to follow for Going-live:

- Procure a Production device via the bank (only for card payments)
- Inform the Ezetap once the device has been procured
- Ezetap team will provide you the Production App Key, add the Production App key in your initialize API request.
- Change the App mode to PROD, from DEMO

Best Practice: Ezetap recommends that you perform some test Re.1 Transactions and verify the transaction flow, portal information etc.