SDK Developer Guide

Project Tracking

KAN Developments
MAY-2021

USER MANUAL
TABLE OF CONTENTS

Page #

1.0 GENERAL INFORMATION

1.1 System Overview

TBD

1.2 Authorized Use Permission

TBD

1.3 Points of Contact

TBD

2.0 SDK FEATURES

2.1 Initialization

At startup of the app fingo sdk needs to be initialized to be able to proceed with api usage.

2.1.1 Fingo SDK Init

1) Provide Context for initializing the FingoSDK as follows:

```
FingoErrorCode fingoInitErrorCode = FingoSDK.initialize(@context);
Log.i(TAG, fingoInitErrorCode);
```

2) After step 1 is finished error code is returned indicating the initialization status
   i. H1_OK if initialization is successful
   ii. Any other error code if initialization failed

2.1.2 Setting FingoParams

1) Create FingoParams object and specify the required parameters

```
FingoParams fingoParams = new FingoParams();
fingoParams.setCloudUrl("cloud_url");
fingoParams.setPartnerId("partner_id");
fingoParams.setMerchantId("merchant_id");
fingoParams.setTerminalId("terminalID");
fingoParams.setApiKey("api_key");
fingoParams.setTemplateKeySeed("template_key_seed");
```

2) Pass the specified Fingoparams to the fingo sdk during to set this params into the fingo SDK.

```
FingoErrorCode fingoParamsErrorCode = FingoSDK.setFingoParams(fingoParams);
```

3) After setting “FingoParams” into the SDK the sdk will return if the params r valid or not in the fingoParamsErrorCode.

2.2 Identification API Reference

Identification api's includes (Enrollment and Identify).

2.2.1 Enrollment

1) User can enroll a finger using the finger's vein ID on the fingo cloud, the cloud will provide a unique deterministic finger vein ID for the finger of choice.

2) During the enrollment process if the user's finger is already enrolled on the cloud, the stored vein ID will be returned immediately in the api response.

3) Enrollment process consists of three finger enroll captures and one verification capture, so total of 4 scans for the enrollment process to be completed.

4) Same Finger should be used during the whole enrollment process.

2.2.2 Identify

1) After a successful enrollment on the fingo cloud, user can be identified by the enrolled finger unique biometric template (vein_id).

2) Identification process consists of one finger capture for the identification process to be completed.

3) If the user is already enrolled on the cloud, his unqiue vein ID will be returned from the cloud, if the user is not enrolled the cloud will return and error. (check the errors section).

2.3 Payment API Reference

Identification api's includes (Enrollment and Identify).

2.3.1 Payment

1) After a successful enrollment on the fingo cloud, user can perform payments on the fingo cloud, given that the user already have a predefined card data stored on the cloud..

2) Payment process consists of one finger capture for the payment process to be completed.

3) If the payment failed the cloud will return error (Refer to payment error section), if the payment is successful the below payment data will be returned from the cloud.

    a Transaction ID

    b Gateway Transaction ID

    c Gateway Auth Code

    d Masked Card Number

    e Merchant ID

    f Transaction Timestamp

2.3.2 Refund

1) After a successful payment on the fingo cloud, user can refund the already made transaction partial refund or full refund, given that the user already have a predefined card data stored and available payment to refund on the cloud.

2) Refund process consists of one finger capture for the refund process to be completed.

3) If the refund failed the cloud will return error (Refer to payment error section), if the refund is successful the below refund data will be returned from the cloud.

    a Transaction ID

    b Gateway Transaction ID

    c Gateway Auth Code

    d Masked Card Number

    e Merchant ID

    f Transaction Timestamp

2.4 Examples

1) First create a FingoListener that will be responsible for the events received from the SDK api's and will provide callbacks with results.

```
FingoContract.FingoListener fingoListener = new
FingoContract.FingoListener() {
    @Override
    public void onProcessingStarted() {

    }

    @Override
    public void onDisplayTextRequested(DisplayTextRequested
displayTextRequested) {

    }

    @Override
    public void onIdentifyData(IdentifyData identifyData) {

    }

    @Override
    public void onPaymentData(PaymentData paymentData,
FingoErrorResponse fingoErrorResponse) {

    }

    @Override
    public void onProcessingFinished(ProcessingFinished
processingFinished) {

    }
}
```

2) Create a presenter that will be responsible to access the SDK api's and will provide callbacks with results

```
fingoPresenter = new FingoPresenter(activity, fingoListener);
```

3) To invoke the enrollment API:

```
fingoPresenter.enroll(TIMEOUT);
```

4) To invoke the identify API:

```
fingoPresenter.identify(TIMEOUT);
```

5) To invoke the payment API:

```
PosData posData = new PosData("2", "SomeLocation");
fingoPresenter.payment(100, Currency.GBP, 0,posData, TIMEOUT);
```

6) To invoke the refund API:

```
TerminalData terminalData = new TerminalData();
terminalData.setLocation("Cairo");
fingoPresenter.refund(100, "8c04ad1b-e1e8-4752-b50c-e3c9dc70ad11",
"96577222", terminalData, TIMEOUT);
```

7) To cancel the finger vein capture session

```
fingoPresenter.cancel();
```

8) To check if the capture session is cancelled by the SDK

```
fingoPresenter.isOperationCancelled()
```

9) For identification and enrollment API's, the results will be received in the

```
onIdentifyData(IdentifyData identifyData)
```

10) For payment and refund API's, the results will be received in the

```
onPaymentData(PaymentData paymentData, FingoErrorResponse
fingoErrorResponse)
```