

Push Payments - Android - Consumer SDK Documentation

- 1 ConsumerSDK-[version].zip
- 2 Steps to Integrate
 - 2.1 A.1.1 API Methods
- 3 A.1.2 API Objects
 - 3.1 A.1.2.1 Pay Merchant Request
 - 3.2 A.1.2.2 Card Details
 - 3.3 A.1.2.3 PayMerchantAliasData
 - 3.4 A.1.2.4 AliasResponse
 - 3.5 QRCodeTag enum
 - 3.5.1 Enum Values
 - 3.6 QRCodeTag.SubTag enum
 - 3.6.1 ENUM VALUES
 - 3.7 A.1.2.4 PayFriendRequest
 - 3.8 A.1.2.5 Payee Object
 - 3.9 PayFriendResponse
 - 3.10 A.1.2.6 MVisaConfig object
 - 3.11 A.1.3 API Interfaces
 - 3.11.1 A.1.3.1 Response Callback Interface
 - 3.11.2 A.1.3.2 AliasValidationHandler
 - 3.11.3 A.1.3.3 PayMerchantAliasResponseHandler
 - 3.12 A.1.3.2 Exceptions
 - 3.13 P2M Alias Code Snippet - Sample

ConsumerSDK-[version].zip

The Android Consumer SDK for push payment include two libraries (one .aar and .jar file) and change log text file.

1. **Consumer SDK** – The Core Consumer SDK library which has all the API that has to be called by the application.
2. **zxing core** – A dependency library for Consumer SDK for QR scanning capability.
3. **Change log text file** - File containing details about the the latest changes.

Steps to Integrate

1. Add the 2 aar files inside the **libs/** folder in the app module of the application project.
2. Add the following block to app module's **build.gradle** file to define the location of the aars.

```
repositories{
    flatDir{
        dirs 'libs'
    }
}
```

3. Add the following lines under dependencies block in the app module's **build.gradle** file to add the visa dependencies

```
compile fileTree(dir: 'libs', include: ['*.jar'])

compile(name: 'ConsumerSDK-2.5.0', ext: 'aar')
```

*Use implementation scope for grade version 3.0 and above

*replace X.X.X with SDK version number

4. Add the following lines under dependencies block in the app module's **build.gradle** to add support library dependencies which the SDK uses.

```
compile 'com.android.support:appcompat-v7:27.1.1'
```

5. Add pro guard rules in obfuscation for release build

```
-keep class com.visa.mvisa.** {
*
}
```

6. Do a gradle sync & build and start using the SDK APIs

```
//Step1: Instantiate SDK

final MVisaSDK mVisaSDK = MVisaSDKImpl.getInstance(getApplicationContext());

//Step2: Configure SDK

mVisaConfig = new MVisaConfig.Builder(getApplicationContext())
    .backgroundColor("#FF4500")
    .actionButtonColor("#B1D9F4")
    .actionButtonActiveColor("#BBD5E5")
    .build();
mVisaSDK.setConfiguration(mVisaConfig);

//Step 3. Make Pay Merchant API call.

/** Make similar calls for payFriend

mVisaSDK.payMerchant(payMerchantRequest, new ResponseCallback<PayMerchantResponse, MVisaErrorObject,
BasePayMerchantActivity>() {
    @Override
    public void onSuccess(final BasePayMerchantActivity activity, PayMerchantResponse response) {
        //handle response
        Toast.makeText(getApplicationContext(), "finished payment", Toast.LENGTH_LONG).show();
        Log.d(TAG, response.toString());
    }

    @Override
    public void onFlowTimeOut(BasePayMerchantActivity activity, final MVisaErrorObject failureObject) {
        //Dont use the method. Will be deprecated
    }
});
```

A.1.1 API Methods

USE CASE	ENTRY POINT METHOD	INPUT PARAMETERS	OUTPUT PARAMETERS
CONSUMER SDK			
Initiate SDK with default Configurations	static MVisaSDK getInstance (Context applicationContext)	Context – android application context	MVisaSDK instance
Initiate SDK with custom Configurations	static MVisaSDK getInstance (Context , MVisaConfig)	Context – android application context MVisaConfig – Configuration Object with custom configurations set	MVisaSDK instance

Click on Pay merchant button	<pre>void payMerchant (PayMerchantRequest, ResponseCallback<PayMerchantRe sponse, MVisaErrorObject, BasePayMerchantActivity>))</pre>	<p>PayMerchantRequest – Request Object with all the Input Parameters for Pay merchant flow</p> <p>ResponseCallback –</p> <p>Implementation of the callback interface for SDK to return back Response object/Error</p>	<p><u>Success callback:</u></p> <p>BasePayMerchantActivity – Instance for the base activity of SDK. User can use this object for closing the SDK screens and navigating to other screens</p> <p>PayMerchantResposne –</p> <p>Response Object for Pay Merchant Flow containing all the information needed for Pay to Merchant transaction.</p> <p><u>Flow Time Out callback:</u></p> <p>BasePayMerchantActivity – Instance for the base activity of SDK. User can use this object for closing the SDK screens and navigating to other screens</p> <p>MVisaErrorObject – Error Object containing error code and error message</p>
Click on Pay merchant button (App supports Visa Aliasing)	<pre>void payMerchantUsingAlias (PayMerchantRequest payMerchant Request, ResponseCallback<PayMerchantRe sponse, MVisaErrorObject, BasePayMerchantActivity> responseCallback, AliasValidationHandler<PayMerchant AliasData, BasePayMerchantActivity> aliasValidationHandler)</pre>	<p>PayMerchantRequest – Request Object with all the Input Parameters for Pay merchant flow</p> <p>ResponseCallback –</p> <p>Implementation of the callback interface for SDK to return back Response object/Error end of the flow</p> <p>AliasValidationHandler –</p> <p>Implementation of callback interface for SDK to pass the Alias data to application for resolving the alias</p>	<p><u>Response Callback</u></p> <p><u>Success callback:</u></p> <p>BasePayMerchantActivity – Instance for the base activity of SDK. User can use this object for closing the SDK screens and navigating to other screens</p> <p>PayMerchantResposne –</p> <p>Response Object for Pay Merchant Flow containing all the information needed for Pay to Merchant transaction.</p> <p><u>Flow Time Out callback:</u></p> <p>BasePayMerchantActivity – Instance for the base activity of SDK. User can use this object for closing the SDK screens and navigating to other screens</p> <p>MVisaErrorObject – Error Object containing error code and error message</p> <p><u>Alias Validation Handler</u></p> <p>doAliasValidation callback:</p> <p>The mobile application should implement this method to make API call to resolve the alias and return the control/AliasResponse back to SDK using PayMerchantAliasResponseHandler parameter passed to this method. This method will get the Alias Data input on UI as PayMerchantAliasData</p>

Click on Pay A Friend Button	void payFriend(PayFriendRequest, ResponseCallback<PayFriendResponse, MVisaErrorObject, BasePayFriendActivity>))	<p>PayFriendRequest – Request Object with all the Input Parameters for Pay Friend flow</p> <p>ResponseCallback –</p> <p>Implementation of the callback interface for SDK to return back Response object/Error</p>	<p>Success Callback:</p> <p>BasePayFriendActivity – Instance for the base activity of SDK. User can use this object for closing the SDK screens and navigating to other screens</p> <p>PayFriendResponse –</p> <p>Response Object for Pay Friend Flow containing all the information needed for Pay to Friend transaction.</p> <p>Flow Time Out:</p> <p>BasePayFriendActivity – Instance for the base activity of SDK. User can use this object for closing the SDK screens and navigating to other screens</p> <p>MVisaErrorObject – Error Object containing error code and error message</p> <p>-</p>
Set configurations (Can also be done when initiating the SDK)	void setConfiguration(MVisaConfig)	MVisaConfig – Configuration object with custom configurations set	

A.1.2 API Objects

The following describes the objects used by the methods in the SDK with their type, purposes and where they are used.

A.1.2.1 Pay Merchant Request

ITEM	DESCRIPTION	FORMAT	METHODS
<p>ArrayList<CardDetails> cardDetails</p> <p>(Note : Please click CardDetails to check the documentation for CardDetails Object)</p>	List of card details object containing consumer card details for payment. Depending on the input SDK will show the cards in Change Card selection screen	<p>MANDATORY</p> <p>Cannot be empty</p>	<p>Setter :</p> <p>setCardDetails (ArrayList<CardDetails> cardDetails)</p> <p>Getter :</p> <p>getCardDetails()</p>
int defaultCardIndex	Index of the default card from the list card details passed. Index starts from 0 for the first card in the list.	<p>MANDATORY</p> <p>Index >= 0</p> <p>Index < size of card details list</p>	<p>Setter :</p> <p>setDefaultCardIndex (int index)</p> <p>Getter :</p> <p>getDefaultCardIndex()</p>
String currencyCode	currencyCode	<p>MANDATORY</p> <p>The NUM column from https://en.wikipedia.org/wiki/ISO_4217_-_Active_codes</p>	<p>Setter :</p> <p>setCurrencyCode(String code)</p> <p>Getter :</p> <p>getCurrencyCode()</p>

A.1.2.2 Card Details

Object to hold the card details to be shown in change card screen

ITEM	DESCRIPTION	FORMAT	METHODS
String cardNickName	Any string which the issuer wants to show before the last four digits in Review Payment Screen. Ex: NICKNAME...xxxx	MANDATORY	Setter: setCardNickName(String nickName) Getter: String getCardNickName
String cardNumberLastFour	Last 4 digits of the card. Will be shown in review payment screen	MANDATORY	Setter: setCardNumberLastFour (String cardNumber) Getter: String getCardNumberLastFour()
int cardArtColor	Color-int for the card art background color	OPTIONAL	Setter: setCardArtColor(String cardArtColorID) Getter Int getCardArtColor()
Drawable issuerLogo	Drawable image of the issuerLogo shown on the top left corner of the cardArt	OPTIONAL	Setter' setIssuerLogo(String issuerLogo) Getter: Drawable getIssuerLogo()
Drawable cartArtImageLayer	Drawable image to fill the background of the cardArt. It can be a transparent texture on top of the color, or opaque which will fill the whole Background of the card	OPTIONAL	Setter: setCardArtImageLayer (String cardArtBackgroundImage) Getter: Drawable getCardArtImageLayer()
Drawable networkLogo	Drawable image to show the network logo for the card	OPTIONAL	Setter : setNetworkLogo(Drawable networkLogo) Getter: Drawable getNetworkLogo()

A.1.2.3 PayMerchantAliasData

ITEM	DESCRIPTION	FORMAT	METHODS
String aliasData	The Alias Data which is input on the UI which will be sent to mobile app using AliasValidationHandler's doAliasValidation method	MANDATORY	Getter : String getAliasData()

A.1.2.4 AliasResponse

ITEM	DESCRIPTION	FORMAT	METHODS
------	-------------	--------	---------

String merchantName	<p>Resolved merchant name for the Alias sent to merchant app.</p> <p>Use this variable to send the merchant name back to SDK using PayMerchantAliasResponseHandler's success method callback</p>	OPTIONAL	<p>Setter :</p> <p>void setName (Sting merchantName)</p>
String merchantCity	<p>Resolved merchant city for the Alias sent to merchant app.</p> <p>Use this variable to send the merchant city back to SDK using PayMerchantAliasResponseHandler's success method callback</p>	MANDATORY	<p>Setter :</p> <p>void setCity (Sting merchantCity)</p>
String recipientPrimaryAccountNumber	<p>Resolved merchant recipientPrimaryAccountNumber for the Alias sent to merchant app.</p> <p>Use this variable to send the merchant recipientPrimaryAccountN umber back to SDK using PayMerchantAliasResponseHandler's success method callback</p>	OPTIONAL	<p>Setter :</p> <p>void setRecipientPrimaryAccountNu mber (recipientPrimaryAccountNumb er)</p>
String issuerName	<p>Resolved merchant issuer name for the Alias sent to merchant app.</p> <p>Use this variable to send the merchant issuer name back to SDK using PayMerchantAliasResponseHandler's success method callback</p>	OPTIONAL	<p>Setter :</p> <p>void setIssuerName (issuerName)</p>
String cardType	<p>Resolved merchant card type (e.g.: VISA PLATINUM) for the Alias sent to merchant app.</p> <p>Use this variable to send the merchant card type back to SDK using PayMerchantAliasResponseHandler's success method callback</p>	OPTIONAL	<p>Setter :</p> <p>void setCardType (cardType)</p>
String postalCode	<p>Resolved merchant postalCode for the Alias sent to merchant app.</p> <p>Use this variable to send the merchant postalCode back to SDK using PayMerchantAliasResponseHandler's success method callback</p>	OPTIONAL	<p>Setter :</p> <p>void setPostalCode (postalCode)</p>
String country	<p>Resolved merchant country for the Alias sent to merchant app.</p> <p>Use this variable to send the merchant country back to SDK using PayMerchantAliasResponseHandler's success method callback</p>	OPTIONAL	<p>Setter :</p> <p>void setCountry (country)</p>
String merchantCategoryCode	<p>Resolved merchant merchantCatego ryCode for the Alias sent to merchant app.</p> <p>Use this variable to send the merchant merchantCategoryCode back to SDK using PayMerchantAliasResponseHandler's success method callback</p>	OPTIONAL	<p>Setter :</p> <p>void setMerchantCategoryCode (mer chantCategoryCode)</p>

String pointOfInitiationMethod	Resolved merchant pointOfInitiationMethod for the Alias sent to merchant app. Use this variable to send the merchant pointOfInitiationMethod back to SDK using PayMerchantAliasResponseHandler's success method callback	OPTIONAL	Setter : void setPointOfInitiationMethod (pointOfInitiationMethod)
String tipAndConvenienceFeeIndicator	Resolved merchant tipAndConvenienceFeeIndicator (e.g.: 01, 02, 03) for the Alias sent to merchant app. Use this variable to send the merchant tipAndConvenienceFeeIndicator back to SDK using PayMerchantAliasResponseHandler's success method callback 01 - Amount and convenienceFee both has to be enter manually 02 - Amount is taken from Alias and convenienceFee can be entered manually 03 - Amount has to be entered manually and convenienceFee is taken from alias.	OPTIONAL	Setter : void setTipAndConvenienceFeeIndicator (tipAndConvenienceFeeIndicator)
String transactionCurrencyCode	Resolved merchant transactionCurrencyCode (accepts numeric/alpha currency code) for the Alias sent to merchant app. Use this variable to send the merchant transactionCurrencyCode back to SDK using PayMerchantAliasResponseHandler's success method callback	OPTIONAL	Setter : void setTransactionCurrencyCode (transactionCurrencyCode)
String convenienceFeeAmount	Resolved merchant convenienceFeeAmount for the Alias sent to merchant app. Use this variable to send the merchant convenienceFeeAmount back to SDK using PayMerchantAliasResponseHandler's success method callback	OPTIONAL	Setter : void setConvenienceFeeAmount (convenienceFeeAmount)
String convenienceFeePercentage	Resolved merchant convenienceFeePercentage for the Alias sent to merchant app. Use this variable to send the merchant convenienceFeePercentage back to SDK using PayMerchantAliasResponseHandler's success method callback	OPTIONAL	Setter : void setConvenienceFeePercentage (convenienceFeePercentage)

A.1.2.5 Pay Merchant Response

PLEASE USE HASHMAP BASED RESPONSE. ALL OTHER VARIABLES IN THIS OBJECT ARE DEPRECATED.

ITEM	DESCRIPTION	FORMAT	METHODS
HashMap<String, Object> mVisaTlvHashMap;	HashMap containing all responses in key value pair. Please use HashMap based response. Use the QrCodeTag enum tagName method as key to retrieve data items	MANDATORY	Getter: HashMap<String, Object> getmVisaTlvHashMap()

String aliasData	Input aliasData. Mobile application should cache the resolved information for aliasData for processing	CONDITIONAL	Getter: String getAliasData()
String payloadFormatIndicator	Defines the format of the merchant data payload.	CONDITIONAL Will be null if Qr Code version is 00	Getter: String getPayloadFormatIndicator()
String pointOfInitiation	Indicates the method by which the data is presented by the merchant. Indicates whether the data is static or dynamic. Refer to version 01 QR Spec	Optional	Getter: String getPointOfInitiation()
String mVisaMerchantId	Deprecated variable, please use getmVisaMerchantPan	MANDATORY	Getter: String getmVisaMerchantPan ()
String mVisaMerchantPan	Merchant Id converted to merchant Pan(16 digit) – should be used in mVisa transaction as merchant id	MANDATORY	Getter: String getmVisaMerchantPan()
String masterCardPan1	Master card pan 1	OPTIONAL	Getter: String getMasterCardPan1()
String masterCardPan2	Master card pan 2	OPTIONAL	Getter: String getMasterCardPan2()
String npciid1	NPCI ID 1	OPTIONAL	Getter: String getNpciid1()
String npciid2	NPCI ID 2	OPTIONAL	Getter: String getNpciid2()
String merchantCategoryCode	Merchant category code as defined in As defined by ISO 8583 for Card Acceptor Business Code.	MANDATORY	Getter: String getMerchantCategoryCode()
String currencyCode	Transaction Currency code as defined by ISO 4217	MANDATORY	Getter : String getCurrencyCode()
String transactionAmount	Transaction Amount	Optional	Getter: String getTransactionAmount()
String additionalAmount	Addition amount (Tip/ convenience fee etc.)	Optional	Getter: String getAdditionalAmount()
String primaryId	Primary Id value (Additional Data - Value 1 as for use as "purchaseIdentifier" in Visa's merchant push payment API)	OPTIONAL	Getter: String getPrimaryId()
String secondaryId	Secondary Id value (Additional Data - Value 2 as for use as "secondaryId" in Visa's merchant push payment API)	OPTIONAL	Getter: String getSecondaryId()
String convenienceFeePercentage	Convenience Fee Percentage	OPTIONAL	Getter: String getConvenienceFeePercentage()
String countryCode	Country code. As defined by ISO 3166.	OPTIONAL	Getter: String getCountryCode()

String merchantName	Merchant Name	OPTIONAL	Getter: String getMerchantName()
String cityName	City Name	OPTIONAL	Getter: String getCityName()
String postalCode	Postal code	OPTIONAL	Getter: String getPostalCode()
String billId	Bill number (Part of Additional Data)	OPTIONAL	Getter: String getAdditionalData()
String mobileNumber	Mobile Number	OPTIONAL	Getter: String getMobileNumber()
String storeId	Store ID	OPTIONAL	Getter: String getStoreId()
String loyaltyNumber	Loyalty number1	OPTIONAL	Getter: String getLoyaltyNumber()
String referenceId	Reference ID	OPTIONAL	Getter: String getReferenceId()
String consumerId	Consumer ID	OPTIONAL	Getter: String getConsumerId()
String terminalId	Terminal ID	OPTIONAL	Getter: String getTerminalId()
String purpose	Purpose	OPTIONAL	Getter: String getPurpose()
String additionalConsumerDataRequest	Additional Consumer Data Request	OPTIONAL	Getter: String getAdditionalConsumerDataRequest()
String addDataMasterCard1	Additional data for Master card 1	OPTIONAL	Getter: String getAddDataMasterCard1()
String addDataMasterCard2	Additional data for Master card 2	OPTIONAL	Getter: String getAdditionalDataMasterCard2()
String addDataNpci1	Additional data for Npci 1	OPTIONAL	Getter: String getAddDataNpci1()
String addDataNpci2	Additional data for Npci 2	OPTIONAL	Getter: String getAddDataNpci2()
String crc	Cyclic Redundant Check value	MANDATORY	Getter: String getCrc()

QRCodeTag enum

This enum represent the key for HashMap response which you receive. It has two methods tagCode() and tagName()

ITEM	DESCRIPTION	Getter Method
------	-------------	---------------

String tagCode	Returns the numerical tagCode which should be used to access an item from the HashMap response	tagCode()
String tagName	Returns the tagName which can be use to access an item from the JSON response	tagName()

Enum Values

```
//ENUM("tagName", "tagCode");

PAYLOAD_FORMAT_INDICATOR("payloadFormatIndicator", "00"),

POINT_OF_INITIATION("pointOfInitiation", "01"),

MERCHANT_ID("mVisaMerchantId", "02"),

MERCHANT_PAN("mVisaMerchantPan", "02"),

ALIAS_ID("aliasId", "03"),)

MASTER_CARD_PAN_1("masterCardPan1", "04"),

MASTER_CARD_PAN_2("masterCardPan2", "05"),

NPCIID1("npciid1", "06"),

NPCIID2("npciid2", "07"),

TAG_08("tag08", "08"),

DISCOVER_1("discover1", "09"),

DISCOVER_2("discover2", "10"),

AMEX_1("amex1", "11"),

AMEX_2("amex2", "12"),

JCB_1("jcb1", "13"),

JCB_2("jcb2", "14"),

UNION_PAY_1("unionPay1", "15"),

UNION_PAY_2("unionPay2", "16"),

TAG_17("tag17", "17"),

TAG_18("tag18", "18"),

TAG_19("tag19", "19"),

TAG_20("tag20", "20"),

TAG_21("tag21", "21"),

TAG_22("tag22", "22"),

TAG_23("tag23", "23"),

TAG_24("tag24", "24"),

TAG_25("tag25", "25"),

TAG_26("tag26", "26"),
```

TAG_27("tag27", "27"),
TAG_28("tag28", "28"),
TAG_29("tag29", "29"),
TAG_30("tag30", "30"),
TAG_31("tag31", "31"),
TAG_32("tag32", "32"),
TAG_33("tag33", "33"),
TAG_34("tag34", "34"),
TAG_35("tag35", "35"),
TAG_36("tag36", "36"),
TAG_37("tag37", "37"),
TAG_38("tag38", "38"),
TAG_39("tag39", "39"),
TAG_40("tag40", "40"),
TAG_41("tag41", "41"),
TAG_42("tag42", "42"),
TAG_43("tag43", "43"),
TAG_44("tag44", "44"),
TAG_45("tag45", "45"),
TAG_46("tag46", "46"),
TAG_47("tag47", "47"),
TAG_48("tag48", "48"),
TAG_49("tag49", "49"),
TAG_50("tag50", "50"),
TAG_51("tag51", "51"),
MERCHANT_CATEGORY_CODE("merchantCategoryCode", "52"),
CURRENCY_CODE("currencyCode", "53"),
TRANSACTION_AMOUNT("transactionAmount", "54"),
CONVENIENCE_FEE_INDICATOR("convenienceFeeIndicator", "55"),
CONVENIENCE_FEE_AMOUNT("convenienceFeeAmount", "56"),
CONVENIENCE_FEE_PERCENTAGE("convenienceFeePercentage", "57"),
COUNTRY_CODE("countryCode", "58"),
MERCHANT_NAME("merchantName", "59"),
CITY_NAME("cityName", "60"),
POSTAL_CODE("postalCode", "61"),
ADDITIONAL_DATA_FIELD_TEMPLATE("additionalDataFieldTemplate", "62"),
CRC("crc", "63"),
MERCHANT_INFORMATION_LANGUAGE_TEMPLATE("merchantInformationLanguageTemplate", "64"),
TAG_65("tag65", "65"),

```
TAG_66("tag66", "66"),
TAG_67("tag67", "67"),
TAG_68("tag68", "68"),
TAG_69("tag69", "69"),
TAG_70("tag70", "70"),
TAG_71("tag71", "71"),
TAG_72("tag72", "72"),
TAG_73("tag73", "73"),
TAG_74("tag74", "74"),
TAG_75("tag75", "75"),
TAG_76("tag76", "76"),
TAG_77("tag77", "77"),
TAG_78("tag78", "78"),
TAG_79("tag79", "79"),
TAG_80("tag80", "80"),
TAG_81("tag81", "81"),
TAG_82("tag82", "82"),
TAG_83("tag83", "83"),
TAG_84("tag84", "84"),
TAG_85("tag85", "85"),
TAG_86("tag86", "86"),
TAG_87("tag87", "87"),
TAG_88("tag88", "88"),
TAG_89("tag89", "89"),
TAG_90("tag90", "90"),
TAG_91("tag91", "91"),
TAG_92("tag92", "92"),
TAG_93("tag93", "93"),
TAG_94("tag94", "94"),
TAG_95("tag95", "95"),
TAG_96("tag96", "96"),
TAG_97("tag97", "97"),
TAG_98("tag98", "98"),
TAG_99("tag99", "99");
```

QRCodeTag.SubTag enum

This enum is used to access subtag elements in the HashMap response

ITEM	DESCRIPTION	Getter Method
String tagCode	Returns the numerical tagCode which should be used to access an item from the HashMap response	tagCode()
String tagName	Returns the tagName which can be use to access an item from the JSON response	tagName()

ENUM VALUES

SUB_TAG_00("subTag00", "00"),

ALTERNATE_LANGUAGE("alternateLanguage", "00"),

SUB_TAG_01("subTag01", "01"),

MERCHANT_NAME_ALTERNATE_LANGUAGE("merchantNameAlternateLanguage", "01"),

BILL_ID("billID", "01"),

SUB_TAG_02("subTag01", "02"),

MERCHANT_CITY_ALTERNATE_LANGUAGE("cityAlternateLanguage", "02"),

MOBILE_NUMBER("mobileNumber", "02"),

SUB_TAG_03("subTag03", "03"),

STORE_ID("storeID", "03"),

SUB_TAG_04("subTag04", "04"),

LOYALTY_NUMBER("loyaltyNumber", "04"),

SUB_TAG_05("subTag05", "05"),

REFERENCE_ID("referenceID", "05"),

SUB_TAG_06("subTag06", "06"),

CONSUMER_ID("consumerID", "06"),

SUB_TAG_07("subTag07", "07"),

TERMINAL_ID("terminalID", "07"),

SUB_TAG_08("subTag08", "08"),

PURPOSE("purpose", "08"),

SUB_TAG_09("subTag09", "09"),

ADDITIONAL_CONSUMER_DATA_REQUEST("additionalConsumerDataRequest", "09"),

SUB_TAG_10("subTag10", "10"),

SUB_TAG_11("subTag11", "11"),

ADD_DATA_MASTER_CARD_1("addDataMasterCard1", "11"),

SUB_TAG_12("subTag12", "12"),

ADD_DATA_MASTER_CARD_2("addDataMasterCard2", "12"),

SUB_TAG_13("subTag13", "13"),

ADD_DATA_NPC1_1("addDataNpci1", "13"),

SUB_TAG_14("subTag14", "14"),

ADD_DATA_NPC1_2("addDataNpci2", "14"),

SUB_TAG_15("subTag15", "15"),

SUB_TAG_16("subTag16", "16"),

SUB_TAG_17("subTag17", "17"),

SUB_TAG_18("subTag18", "18"),

SUB_TAG_19("subTag19", "19"),

SUB_TAG_20("subTag20", "20"),

SUB_TAG_21("subTag21", "21"),

SUB_TAG_22("subTag22", "22"),

SUB_TAG_23("subTag23", "23"),

SUB_TAG_24("subTag24", "24"),

SUB_TAG_25("subTag25", "25"),

SUB_TAG_26("subTag26", "26"),

SUB_TAG_27("subTag27", "27"),

SUB_TAG_28("subTag28", "28"),

SUB_TAG_29("subTag29", "29"),

SUB_TAG_30("subTag30", "30"),

SUB_TAG_31("subTag31", "31"),

SUB_TAG_32("subTag32", "32"),

SUB_TAG_33("subTag33", "33"),

SUB_TAG_34("subTag34", "34"),

SUB_TAG_35("subTag35", "35"),

SUB_TAG_36("subTag36", "36"),

SUB_TAG_37("subTag37", "37"),

SUB_TAG_38("subTag38", "38"),

SUB_TAG_39("subTag39", "39"),

SUB_TAG_40("subTag40", "40"),
SUB_TAG_41("subTag41", "41"),
SUB_TAG_42("subTag42", "42"),
SUB_TAG_43("subTag43", "43"),
SUB_TAG_44("subTag44", "44"),
SUB_TAG_45("subTag45", "45"),
SUB_TAG_46("subTag46", "46"),
SUB_TAG_47("subTag47", "47"),
SUB_TAG_48("subTag48", "48"),
SUB_TAG_49("subTag49", "49"),
SUB_TAG_50("subTag50", "50"),
SUB_TAG_51("subTag51", "51"),
SUB_TAG_52("subTag52", "52"),
SUB_TAG_53("subTag53", "53"),
SUB_TAG_54("subTag54", "54"),
SUB_TAG_55("subTag55", "55"),
SUB_TAG_56("subTag56", "56"),
SUB_TAG_57("subTag57", "57"),
SUB_TAG_58("subTag58", "58"),
SUB_TAG_59("subTag59", "59"),
SUB_TAG_60("subTag60", "60"),
SUB_TAG_61("subTag61", "61"),
SUB_TAG_62("subTag62", "62"),
SUB_TAG_63("subTag63", "63"),
SUB_TAG_64("subTag64", "64"),
SUB_TAG_65("subTag65", "65"),
SUB_TAG_66("subTag66", "66"),
SUB_TAG_67("subTag67", "67"),
SUB_TAG_68("subTag68", "68"),
SUB_TAG_69("subTag69", "69"),
SUB_TAG_70("subTag70", "70"),
SUB_TAG_71("subTag71", "71"),
SUB_TAG_72("subTag72", "72"),
SUB_TAG_73("subTag73", "73"),
SUB_TAG_74("subTag74", "74"),
SUB_TAG_75("subTag75", "75"),
SUB_TAG_76("subTag76", "76"),
SUB_TAG_77("subTag77", "77"),
SUB_TAG_78("subTag78", "78"),

```

SUB_TAG_79("subTag79", "79"),
SUB_TAG_80("subTag80", "80"),
SUB_TAG_81("subTag81", "81"),
SUB_TAG_82("subTag82", "82"),
SUB_TAG_83("subTag83", "83"),
SUB_TAG_84("subTag84", "84"),
SUB_TAG_85("subTag85", "85"),
SUB_TAG_86("subTag86", "86"),
SUB_TAG_87("subTag87", "87"),
SUB_TAG_88("subTag88", "88"),
SUB_TAG_89("subTag89", "89"),
SUB_TAG_90("subTag90", "90"),
SUB_TAG_91("subTag91", "91"),
SUB_TAG_92("subTag92", "92"),
SUB_TAG_93("subTag93", "93"),
SUB_TAG_94("subTag94", "94"),
SUB_TAG_95("subTag95", "95"),
SUB_TAG_96("subTag96", "96"),
SUB_TAG_97("subTag97", "97"),
SUB_TAG_98("subTag98", "98"),
SUB_TAG_99("subTag99", "99");

```

A.1.2.4 PayFriendRequest

ITEM	DESCRIPTION	FORMAT	METHODS
ArrayList<CardDetails> cardDetails (Note : Please click CardDetails to check the documentation for CardDetails Object)	List of card details object containing consumer card details for payment. Depending on the input SDK will show the cards in Change Card selection screen	MANDATORY Cannot be empty	Setter : setCardDetails (ArrayList<CardDetails>) Getter : getCardDetails()
int defaultSenderCardIndex	Index of the default card from the list card details passed to CardDetails. Index starts from 0 for the first card in the list.	MANDATORY Index >= 0 Index < size of card details list	Setter : setDefaultCardIndex (int index) Getter : getDefaultCardIndex()
String currencyCode	currencyCode	MANDATORY The NUM column from https://en.wikipedia.org/wiki/ISO_4217_-_Active_codes	Setter : setCurrencyCode(String code) Getter : getCurrencyCode()

ArrayList<Payee> payees	List of Payee details. User will be able to pick of these payees for P2P transfer or add a new Payee	OPTIONAL	Setter setPayees(ArrayList<Payee> payee) Getter : getPayee()
-------------------------	--	----------	---

A.1.2.5 Payee Object

ITEM	DESCRIPTION	FORMAT	METHODS
String payeeName	Name of the Payee To be shown in the Payee details screen	MANDATORY	Setter: setPayeeName(String Boolean) Getter: String getPayeeName
String payeeCardNumber	Last 4 digits of the card. Will be shown in Payee Details screen	MANDATORY	Setter: setPayeeCardNumber (String cardNumber) Getter: String getPayeeCardNumber ()
Boolean aliasFlow	Flag to switch the UI flow to show only Confirm Payment screen post alias resolving. True for aliasing flow and false for usual flow(usual flow starts from payee screen)	OPTIONAL	Setter: setAliasFlow(Boolean aliasFlow)
AliasPayType paymentType	Enum for specifying what type alias confirm payment screen. 1. NONE – Non Alias flow with PAN 2. Alias – Alias entry present in VAD. Alias will be shown instead of PAN in cnfirm payment screen UI 3. CardFallback – Alias entry not found in VAD. Manual entry of PAN	OPTIONAL	Setter: setPaymentType(AliasPayType paymentType)

PayFriendResponse

ITEM	DESCRIPTION	METHODS
String payeeName	Payee name for making P2P transaction	String getPayeeName()
String payeeCardNumber	16 digit card number of the payee for making P2P transaction	String getPayeeCardNumber()
boolean newPayee	Boolean flag which indicates if the payee is a new payee or not	boolean isNewPayee()
boolean shouldAddNewPayeeToPayeeList	Boolean flag which indicated if the newPayee should be added to payeeList. Valid only if newPayee is truw	boolean isShouldAddNewUserToPayeeList()
Int payeeIndex	Index of the selected payee from the arraylist of payee sent in request	Int getPayeeIndex()
Int senderCardIndex	Index of the selected card used for payment from the arraylist of carddetails sent in request	Int getSelectedCardIndex()

String payeeMobileNumber	Alias Data	String getPayeeMobileNumber()
String aliasPayType	aliasPayType	String getAliasPayType()

A.1.2.6 MVisaConfig object

The MVisa Config object can be built using `MVisaConfig.Builder builder = new MVisaConfig.Builder();`

All the different configurations exposed to the client app can be configured using builder object.

It is not necessary to add all the configurations. Only the configurations needed can be added. Rest of the configuration will take the default values OR previously set value if any.

Note: The MVisaConfig object is a different object for Consumer SDK and Merchant SDK

Package name for Consumer SDK - `com.visa.mvisa.sdk.models.facade.MVisaConfig`

Package name for Merchant SDK - `com.visa.mvisa.merchantsdk.models`

`.facade.MVisaConfig`

USE CASE	ENTRY POINT METHOD	INPUT PARAMETERS (& default values if any)	OUTPUT PARAMETERS	Flow
Configure toggling on/off to show/hide NFC tab in P2M scan QR flow	MVisaConfig.Builder isNfcOptionOn(boolean)	True – to toggle on NFC tab in P2M flow; False- to toggle off the NFC tab in P2P flow Default - true	The builder object with all the configs added so far, call build using this object to build MVisaConfig object	Pay Merchant
Configure the logo icon in the android action bar left corner instead of the default back button	MVisaConfig.Builder issuerBackButton(Drawable icon)	Icon – icon to show in the header in place of back button as a drawable (Preference is BitmapDrawable) Default – default back icon as per design wireframes	The builder object with all the configs added so far, call build using this object to build MVisaConfig object	ALL
Set the dark background color in the SDK screens	MVisaConfig.Builder backgroundColor(String hexColor)	Hexadecimal color for background Default - #FF15195A	The builder object with all the configs added so far, call build using this object to build MVisaConfig object	ALL
Set the dark action button background color in the SDK screens	MVisaConfig.Builder actionButtonColor (String hexColor)	Hexadecimal color for action button background Default - #FF003EA9	The builder object with all the configs added so far, call build using this object to build MVisaConfig object	ALL
Set the pressed state color for the action button in the sdk screens	MVisaConfig.Builder actionButtonActiveColor (String hexColor)	Hexadecimal color for action button pressed state color Default - #FF1869D6	The builder object with all the configs added so far, call build using this object to build MVisaConfig object	ALL
Set the font color for the text on top of dark background	MVisaConfig.Builder fontColor (String hexColor)	Hexadecimal color for font color to be shown on top of background color Default -#FFFFFF	The builder object with all the configs added so far, call build using this object to build MVisaConfig object	ALL

Set the font color for the text on top of action buttons	MVisaConfig.Builder actionButtonFontColor (String hexColor)	Hexadecimal color for font color to be shown on top of action button font color Default -#FFFFFF	The builder object with all the configs added so far, call build using this object to build MVisaConfig object	ALL
Set the network logo list for network branding	MVisaConfig.Builder networkLogoList (ArrayList<Drawable> hexColor)	List if Drawable objects containing the supported network logo branding images (like Visa, mastercard logo etc). This will be shown at the bottom scan page Default – Visa		Pay Merchant
Set the show currency selection	MVisaConfig.Builder showCurrencySelection(boolean showCurrencySelection)	Setting to populate the list of currency in the confirm payment screen. By default it is set to true, in case of alias flow this has to be set to false. As currency is read from Alias response in that case.		
To build the MVisaConfig Object	MvisaConfig build()		MVisaConfig object. This object should be passed when initializing the SDK (getInstance) or using setConfigurations API	ALL

A.1.3 API Interfaces

The following describes the interfaces used in the SDK with their purposes, methods & their descriptions.

A.1.3.1 Response Callback Interface

The application has to implement the ResponseCallback<T,R,A> Interface to listen to exit events fired by the SDK.

T- Type of Response Object

R- Type of errorObject

A- Type of Base Activity

The response callback interface has two methods:

1. 1. **onSuccess(A baseActivityOfTheFlow, T responseObject)**

This callback method will be called when the SDK flow ends successfully and the response is sent back to the user.

1. 2. **onFlowTimeOut(A baseActivityOfTheFlow, R errorObject) - Deprecate ** DO NOT USE TIMEOUT METHODS ****

This callback method will be called when there is no user interaction for 'x' seconds at a particular screen. The value of 'x' variable is configurable. To close the SDK Screens inside this callback, please call baseActivityOfTheFlow.finish(). It is always recommended to call finish() inside onSuccess() and onFlowTimeOut(). There is no flow timeout for card scan page in Pay friend flow.

A.1.3.2 AliasValidationHandler

The Application has to implement the AliasValidationHandler<R, T> interface to get the alias data input on UI for sending it to Visa Aliasing Directory for validation. It has one method doAliasValidation method where the backend implementation to Mobile Application platform to resolve alias data is implemented. The doAliasValidation method gets three input

R – Type of object to get the input Alias Data. (PayMerchantAliasData)

T- Type of Base Activity

1. doAliasValidation(R aliasData, T activity , PayMerchantAliasResponseHandler<AliasResponse> aliasResponseHandler)

The method is implemented by mobile application to resolve alias data. Once resolved, the control/data can be passed back to SDK to continue the UI flow using PayMerchantAliasResponseHandler.

A.1.3.3 PayMerchantAliasResponseHandler

-

SDK will pass an implementation of this method to mobile app to get a call back post alias resolution.

onSuccess(AliasResponse response)

The mobile application is supposed to call the SDK back on successful API call. If there is no alias entry in VAD. The response object should be null. The input will be considered as merchant PAN/merchant ID incase response is null and merchant ID validation will be done on the input.

onFailure(MVisaErrorObject errorObject)

On backend failure or other means of failure, mobile application should call this method with MVisaErrorObject. If the application wants to show a custom error message, it can pass the message through MVisaErrorObject.

A.1.3.2 Exceptions

FLOW	USE CASE	EXCEPTION	MESSAGE
getInstance flow	Input null for context	InputInvalidException	Cannot initialize SDK with null Context. Please pass a valid context
getInstance flow	Input null for MVisaConfig object	InputInvalidException	Cannot initialize SDK with null mVisaConfig. Please pass a valid mVisaConfig or call the getInstance (Context) to initialize with default configuration
Build MVisaConfig	Input null for issuerBackButton	InvalidInputException	issuerBackButton cannot be null
Build MVisaConfig	Input negative value for payMerchantTimeOut in seconds	InvalidInputException	payMerchant TimeoutIn seconds cannot be negative
Build MVisaConfig	Input null or empty for backgroundColor	InvalidInputException	backgroundColor cannot be null or empty
Build MVisaConfig	Input null or empty for actionButtonColor	InvalidInputException	actionButtonColor cannot be null or empty
Build MVisaConfig	Input null or empty for actionButtonActiveColor	InvalidInputException	actionButtonActiveColor cannot be null or empty
Build MVisaConfig	Input null or empty for fontColor	InvalidInputException	fontColor cannot be null or empty
Build MVisaConfig	Input null or empty for actionButtonFontColor	InvalidInputException	actionButtonFontColor cannot be null or empty
Build MVisaConfig	Input Invalid hex color for backgroundColor	InvalidInputException	Unknown color for backgroundColor
Build MVisaConfig	Input Invalid hex color for actionButtonColor	InvalidInputException	Unknown color for actionButtonColor
Build MVisaConfig	Input Invalid hex color for actionButtonActiveColor	InvalidInputException	Unknown color for actionButtonActiveColor
Build MVisaConfig	Input Invalid hex color for fontColor	InvalidInputException	Unknown color for fontColor

Build MVisaConfig	Input Invalid hex color for actionButtonFontColor	InvalidInputException	Unknown color for actionButtonFontColor
Build MVisaConfig	call MVisaConfig.Builder constructor with null context	InvalidInputException	Cannot initialize MVisaConfig builder object with null context. Please pass a valid context
PayMerchant Flow	Input null for PayMerchantRequest	InvalidInputException	PayMerchantRequest cannot be null
PayMerchant Flow	Input null or empty for card details arraylist	InvalidInputException	CardDetails arraylist cannot be null or empty
PayMerchant Flow	Input null for any one of the CardDetails object inside the card details array list	InvalidInputException	CardDetails cannot be null
PayMerchant Flow	Input null for cardNickName	InvalidInputException	cardNickName cannot be null
PayMerchant Flow	Input null for cardLastFour	InvalidInputException	cardLastFour cannot be null
PayMerchant Flow	Input invalid length for card last 4	InvalidInputException	cardLastFour should be of length 4
PayMerchant Flow	Input null for currency code string	InvalidInputException	PayMerchantRequest cannot be null
PayMerchant Flow	Input negative value for defaultCardIndex	InvalidInputException	Card index cannot be negative or greater than the CardDetails arraylist size
PayFriend Flow	Input null for PayFriendRequest	InvalidInputException	PayFriendRequest cannot be null
PayFriend Flow	Input null for any one of the CardDetails object inside the card details array list	InvalidInputException	CardDetails cannot be null
PayFriend Flow	Input null for cardNickName	InvalidInputException	cardNickName cannot be null
PayFriend Flow	Input null for cardLastFour	InvalidInputException	cardLastFour cannot be null
PayFriend Flow	Input invalid length for card last 4	InvalidInputException	cardLastFour should be of length 4
PayFriend Flow	Input null for currency code string	InvalidInputException	PayMerchantRequest cannot be null
PayFriend Flow	Input negative value for defaultSenderCardIndex	InvalidInputException	Card index cannot be negative or greater than the CardDetails arraylist size
PayFriend Flow	Input null for Payee inside Payee Arraylsit	InputInvalidException	Payee cannot be null
PayFriend Flow	Input null for payeeName	InvalidInputException	payeeName cannot be null
PayFriend Flow	Input null for payeeCardNumber	InvalidInputException	payeeCardNumber cannot be null
PayFriend Flow	Input invalid length for payeeCardNumber. Length should be 4	InvalidInputException	payeeCardNumber should be of length 4
Pay Merchant Alias Flow	aliasResponse is null	InvalidInputException	AliasResponse object cannot be null
Pay Merchant Alias Flow	Merchant name is null in alias response	InvalidInputException	Merchant Name cannot be null
Pay Merchant Alias Flow	Merchant name is empty in alias response	InvalidInputException	Merchant Name cannot be empty
Pay Merchant Alias Flow	Merchant name is more than 25 character	InvalidInputException	Merchant Name cannot be more than 25 characters
Pay Merchant Alias Flow	Merchant PAN is null or empty	InvalidInputException	Merchant PAN cannot be null or empty
Pay Merchant Alias Flow	Currency code is null or empty	InvalidInputException	Transaction Currency code cannot be null or empty
Pay Merchant Alias Flow	Currency code length is not 3	InvalidInputException	Transaction Currency code length cannot be other than 3
Pay Merchant Alias Flow	Merchant category code is null or empty	InvalidInputException	Merchant category code cannot be null or empty
Pay Merchant Alias Flow	Merchant category code length is	InvalidInputException	Merchant category code length

	not 4		cannot be other than 4
Pay Merchant Alias Flow	Merchant city is null or empty	InvalidInputException	Merchant City cannot be null or empty
Pay Merchant Alias Flow	Merchant city length is more than 15	InvalidInputException	Merchant City length cannot be more than 15
Pay Merchant Alias Flow	Country code is null or Empty	InvalidInputException	Country code cannot be null or empty
Pay Merchant Alias Flow	country code length is not 2	InvalidInputException	Country code length cannot be other than 2

P2M Alias Code Snippet - Sample

```

AliasValidationHandler aliasValidationHandler = new AliasValidationHandler<PayMerchantAliasData,
BasePayMerchantActivity>() {
    @Override
    public void doAliasValidation(PayMerchantAliasData aliasData, BasePayMerchantActivity activity,
PayMerchantAliasResponseHandler<AliasResponse> aliasCallback) {
        try {
            //alias data input on UI
            String data = aliasData.getAliasData();

            //Implement backend call to resolve alias Data
            //Call success on aliasCallback object which is received from SDK. Call failure in case of backend
            API failure
            AliasResponse aliasResponse = new AliasResponse();
            aliasResponse.setCity("New York");
            aliasResponse.setName("Starbucks Coffee");
            aliasCallback.onSuccess(aliasResponse);

        } catch (InputInvalidException inputInvalidException) {
            Toast.makeText(getApplicationContext(), "InputInvalidException", Toast.LENGTH_LONG).show();
            Log.e(INPUT_INVALID_EXCEPTION_TAG, inputInvalidException.getMessage());
        }
    }
};

mVisaSDK.payMerchantUsingAlias(payMerchantRequest, new ResponseCallback<PayMerchantResponse,
MVisaErrorObject, BasePayMerchantActivity>() {
    @Override
    public void onSuccess(final BasePayMerchantActivity activity, PayMerchantResponse response) {
        //access response object and make backend service call
        Log.d(TAG, response.toString());
        final Intent i = new Intent(activity, SuccessScreen.class);
        i.putExtra("result", response.toString());

    }

    @Override
    public void onFlowTimeOut(BasePayMerchantActivity activity, MVisaErrorObject failureObject) {
        //do not use - deprecated
    }
}, aliasValidationHandler);

```