Corporate Wallet SDK API Specification

Version 2.2.1

11 February 2021

Notices

Following are policies pertaining to proprietary rights and trademarks.

Proprietary Rights

The information contained in this document is proprietary and confidential to Mastercard International Incorporated, one or more of its affiliated entities (collectively “Mastercard”), or both.

This material may not be duplicated, published, or disclosed, in whole or in part, without the prior written permission of Mastercard.

Trademarks

Trademark notices and symbols used in this document reflect the registration status of Mastercard trademarks in the United States. Please consult with the Customer Operations Services team or the Mastercard Law Department for the registration status of particular product, program, or service names outside the United States.

All third-party product and service names are trademarks or registered trademarks of their respective owners.

Disclaimer

Mastercard makes no representations or warranties of any kind, express or implied, with respect to the contents of this document. Without limitation, Mastercard specifically disclaims all representations and warranties with respect to this document and any intellectual property rights subsisting therein or any part thereof, including but not limited to any and all implied warranties of title, non-infringement, or suitability for any purpose (whether or not Mastercard has been advised, has reason to know, or is otherwise in fact aware of any information) or achievement of any particular result. Without limitation, Mastercard specifically disclaims all representations and warranties that any practice or implementation of this document will not infringe any third party patents, copyrights, trade secrets or other rights.

Summary of Changes, 11 February 20211

This document reflects changes associated with ***Corporate Wallet SDK API Specification Specification***.

Description of Change
Below are the changes incorporated in Phase 2.2.1 • Below API added – - digitize (Using Encrypted Card Information) - digitizeByList (Using Encrypted Card Information) • Description & Error Codes section updated in digitizeByList API. • Description & Note section updated in paymentTokenDigitized.

Table of Contents

1 Introduction

1.1 Document Scope

The Corporate Wallet Android SDK will provide the implementation for all of the wallet services as defined in the Functional Specifications document (For latest Functional Specification Document, refer References Section).

The purpose of this document is to identify the APIs and their specifications for Corporate Wallet Android SDK.

1.2 Revision History

Date	Version	Author	Comments
16th Feb, 2016	V0.1	Mastercard	First draft
19 Feb, 2016	V0.5		- Added Check Wallet PIN - Review comments - Merged Wallet PIN authentication with Terminate Wallet, Delete Wallet, get HCE transaction history
22 Feb, 2016	V0.6		- Added “Authorization Token” to Digitized API - Added description note to Reset Wallet PIN API
26 Feb, 2016	V0.7-0.9		- Added Delivery split column - Updated Initialization API - Added check device eligibility - Updated register API - Renamed “Check Wallet Pin” into “Wallet PIN Syntax Check” - Updated “Get Local Card List” - Updated “Notifications from Corporate Wallet SDK to WL-MPA”
04 March, 2016	V0.10 - 0.12		- Reformatted the document with the separate chapter for each APIs. - Description added for each API.
08 March, 2016	V0.13 – 0.14		- Updated diagram and description on solution overview. - Incorporated review comments
09 March 2016	V0.15		- Added setLVTThresholdsAndCurrency - Added transaction object
10 March 2016	V0.16		- Added transaction object - Added transactionAuthorized API - Reviewed Santander comments
20 March 2016	V0.17 – V0.18		- Changes in few of the methods after discovery phase 2 - Added activate, sendActivationCode, getTnC methods
23 March 2016	V0.19		- Added TnC object structure - Cosmetics changes
24 March 2016	V0.20		- Added decisioningData object structure

Date	Versio n	Author	Comments
15 April 2016	V0.21		- Few Modifications post LLD - Modifications in object structure
15 April 2016	V0.22		- Updated section 8.1.1.1 – Added a configuration parameter 'buildType'
22 April 2016	V0.23		- Modified TnC object
27 April 2016	V0.24		- Added further description for buildType parameter in section 8.1.1.1
28 April 2016	V0.25		- Updated getTnC in 4.3 to indicate when it can be called, and to precise stability criteria of the returned TnC.
2 May 2016	V0.26		- Added 'Synchronize' API
11 May 2016	V0.27		- Change of naming from 'Card' to 'PaymentToken' across all the APIs and description. - In naming APIs and parameters, 'Card' is used for non-digitzed Card information - 'PaymentToken' is used for digitized card. Changes made in Following APIs: - Sync API - Added returned parameters - handleNotification (Added New API to handle push messages) - createWallet - Clarification in description - Digitize - Would do digitization of one card in place of list of card. - Changes in input/output params type – - Clarification in description - Change in API name of getTnC to getTermsAndCondition. - getTermsAndCondition does not take any parameters - Change in API name of sendActivationCode to requestActivationCode - Added input parameter activationMethod - activate API - returns activationResult in place CardInfo. - Changes in following APIs name - getLocalCardList to getPaymentTokens - deleteCard to deletePaymentToken - suspendCard to suspendPaymentToken - setDefaultCard to setDefaultPaymentToken - getDefaultCard to getDefaultPaymentToken - replenishTokens to replenishPaymentToken - Change in Notifications - Same notification would be raised for all the PaymentToken state changes. - Added walletUpdated notification - Added paymentTokenUpdated notification - Removed cardSuspended, cardResumed, cardDeleted, cardReadyToPay notifications.

Date	Version	Author	Comments
			- Added Object structures of following – - CardInfo - PaymentToken - SyncResult - WalletInfo - ActivationMethod - ActivationResult -
24 May 2016	V0.28		- Changes in Synchronize API contract - Added isDefault attribute in PaymentToken - Changes in following APIs name - setWalletNotificationHandler to setWalletNotifiacationListener - setTransactionNotifiationHandler to setTransactionNotificationListener.
1 JUN 2016	V0.29		- Changes in Object structure - Removed Card_Type from PaymentToken - Added SecureCode in CardInfo - Changes in API - error code changes in getPaymentTokens
6 JUN 2016	V0.30		- Added Errorcodes for following APIs - activate, digitize, getDefaultPaymentToken, setDefaultPaymentToken, getPaymentToken, requestActivationCode
23 JUN 2016	V0.31		- Changes in APIs related to Transaction process. - Change in params of initiateTransaction - Added callback event getAuthenticationState - Added notification onTransactionStarted - Changes in getTokenForOperation API. - Added params used in L2 authentication for various operations. - Added/Modified Errorcodes of few APIs. - Added information regarding Android Permissions in Appendix
1 JUL 2016	V0.32		- Added Visa component in Fig 1 and description. - Correction from 'TermsAndCondition' to 'TermsAndConditions' at all occurrences. - Added 'INVALID_INPUT' error code - Changes in getAuthenticationState notification. - Added AuthenticationState Object - Removed configuration for AccumulatedAmtThreshold and MinimumLUKThreshold.
5 JUL 2016	V0.33		- Restored configuration for AccumulatedAmtThreshold
8 JUL 2016	V0.34		- Clean-up of comments / revision marks
12 JUL 2016	V0.35		- Added following new APIs/Notifications/Objects - paymentTokenReplenished - TransactionInfo - Changes in following API contracts - SyncResults (added walletInfo)

Date	Version	Author	Comments
			- onTransactionCompleted (passes TransactionInfo as input parameter) - getAuthenticationState(parameter type changed to AuthenticationState) - changes in error codes of synchronize, getTransactionHistory
01 AUG 2016	V0.36		- Added new API "cancelTransaction" - Changes in configuration parameters. - Added few required Wallet Server configuration parameters, TenantId and PaymentAppProviderId. - Changes of error codes in few APIs.
22 AUG 2016	V0.37		- Added Error Description section (8.2) - Added MobilePaymentApplicationId to configuration parameters.
1 SEP 2016	V0.38		- Updated Error Codes and Description in section 4.2, 7.2.5 and 8.2 - Updated android permissions in section (8.3)
2 SEP 2016	V0.39		- Updated Error Code and Description in section 4.4, and 8.2
6 SEP 2016	V0.40		- Added new event getConsent in section 7.2.1 - Updated error code in section 6.1 & 8.2
29 SEP 2016	V0.41		- Added New APIs - registerWithMdes - registerWithVts - getWalletInfo - Modified APIs - getWalletStatus (deprecated this API) - Modified Objects - DecisioningData (Added more attributes) - WalletInfo (Added more attributes) - TransactionDetails (Made merchantPostal code optional)
6 OCT 2016	V0.42		- New Error codes are added for following APIs - registerWithMdes - registerWithVts - digitize - resumePaymentToken - Removed few DecisioningData attributes (those added in V0.41) - Added VTS_Signature_PublicKey in configuration parameters.
18 OCT 2016	V0.43		- Modified APIs - registerWithMdes (Added parameter gcmRegistrationID) - Deprecated MobilePaymentApplicationId from configuration parameters. - Deprecated 'INTERNAL_SYSTEM_ERROR' error code from getDefaultPaymentToken API - Description updated for VTS_Signature_PublicKey in CW-SDK configuration parameters - Description correction for response parameters in performEnvironmentalChecks service

Date	Version	Author	Comments
			- Added new check of Secure Unlock Enabled in performEnvironmentalChecks service Description updated for 'expiryYear' attribute in CardInfo business object.
28 OCT 2016	V0.44		- Added a special case in description of getWalletInfo API - New Error codes are added for following APIs - getTransactionHistory - onTransactionError - digitize - requestActivationCode - activate - deletePaymentToken - suspendPaymentToken - resumePaymentToken - registerWithMdes - registerWithVts - replenishPaymentToken - Error Descriptions section is updated for above error codes - Deprecated Error code FIRST_TAP_TIMEOUT from onTransactionError API - Business Object updates - New object "SuspendedBy" is introduced - New business object EnvironmentCheck is added - New Attribute "suspendedBy" is added in PaymentToken
13 DEC 2016	V0.45		- New Error codes are added for following API - getTokenForOperation - Added a special case in description of getTransactionHistory API - Description updated for following APIs - setDefaultPaymentToken - replenishPaymentToken
21 Feb 2017	V0.46		- Auto Recovery use case specific changes - Added onRecoveryRequired wallet notification API. - Added startRecovery API. - Added getRecoveryStatus API. - Added RecoveryStatus business object - Added reset API. - Added RECOVERY_REQUIRED error code to Multiple APIs like digitize, registerWithMdes, etc. - Digitize multiple cards by list use case specific changes - Added digitizeByList API. - Added DigitizationFailedReason business object. ▪ PaymentToken is updated with digitization failed reason attribute. - performCVM transaction notification API specific changes - Name is corrected to onPerformCVM. - Description is updated for terminal type information - Added business object TransactionContext.

Date	Version	Author	Comments
			- Added Input parameter TransactionContext in onPerformCVM notification API. - **Deprecated** onPerformCVM API (without any input parameters). - createWallet API specific changes - Removed INVALID_WALLET_PIN_FORMAT error code. - Added INVALID_WALLET_STATE error code. - onTransactionError transaction notification specific changes - input parameter name is corrected to TransactionError - Name of the business object Error is corrected to TransactionError - Description of attributes are updated for VelocityCheckRules - Error code AUTH_TOKEN_CREDENTIAL_MISMATCH is added in following APIs & in Error Descriptions - digitize - digitizeByList - registerWithMdes - registerWithVts - startRecovery - Error code DEVICE_NOT_SUPPORTED and OS_NOT_SUPPORTED is added in few APIs like - digitize - digitizeByList - deletePaymentToken - resumePaymentToken etc.
22 March 2017	V0.47		- New Provision Failed Notification is added - New setProvisionFailedNotificationListener API is added
11 July 2017	V0.48		- Description updated for terminateWallet and getWalletInfo API for a helper to know the wallet termination reason. - New Error Code READ_PHONE_STATE_PERMISSION_REQUIRED is added in APIs like - checkDeviceEligibility - createWallet - registerWithMdes - registerWithVts - digitize - digitizeByList etc.
17 November 2017	V2.0.0	Mastercard	- Updated - createWallet (In case of “HCE only”) - Added - performEnvironmentalChecks (New Static API) - createWallet(In case of Masterpass/Converge) - getConsumerProfile - updateConsumerProfile - updateTermsAndConditions - Shipping Address Services - Payment Card Services - W2A Checkout Services - isCheckoutNotification

Date	Version	Author	Comments
			- parseCheckoutNotification - Business Objects
08 December 2017	V2.0.1	Mastercard	- Updated below sections: - Added Error Code **INVALID_CHECKOUT_STATUS** in below APIs: - authorizeCheckout - addShippingAddress (During Checkout) - updateShippingAddress (During Checkout) - Added Error Code **INVALID_OAUTH_TOKEN** in below APIs: - addShippingAddress (During Checkout) - updateShippingAddress (During Checkout) - Added Error Code **MISSING_EXPIRY_DATE** in below APIs: - addPaymentCards - updatePaymentCard - Added Error Code **PAYMENT_CARD_ALREADY_EXISTS** in below API: - addPaymentCards - Removed Error Code **PAYMENT_CARD_ALREADY_ADDED** in below API: - addPaymentCards - Removed Error Code **OPERATION_NOT_SUPPORTED** in below API: - createWallet (In case of Masterpass/Converged) - Below API names have been changed: - retrieveShippingAddresses to getShippingAddress - retrievePaymentCards to getPaymentCards - Below Business Objects have been updated: - PaymentCardInfo - CardDetails - AddPaymentCardsResult - Removed below APIs: - performEnvironmentalChecks - getWalletStatus - onPerformCVM (Older Version) - createWallet (In case of "HCE only") - Removed **MobilePaymentApplicationId** from Application wide configurations table.
15 February 2018	V2.0.2	Mastercard	Below are the Phase 2.0 SIT R3 changes: - Updated below sections: - Added acceptSafetyNetFailedRisk under Wallet Services section. - Added safetyNetFailed under Wallet Notifications section. - Added Google API key under CW-SDK Configuration Parameters and Preferences section. - Updated addPaymentCards, updatePaymentCard, and deletePaymentCard APIs.

Date	Version	Author	Comments
			- Added Error Code **CANNOT_DELETE_LAST_SHIPPING_ADDRESS** in below API: - deleteShippingAddress - Added Error Codes **INVALID_BILLING_ADDRESS** and **INVALID_CHECKOUT_STATUS** in below API: - authorizeCheckout - Added Error Code **USER_ALREADY_EXIST** in below APIs: - updateConsumerProfile and - createWallet (Masterpass/Converged) - Added UpdatePaymentCardInfo in Business Objects.
19 March 2018	2.1.0	Mastercard	Below are the changes incorporated in Phase 2.1 Sprint 2: - Updated certificate related changes in Server Host configurations section - Error Codes section removed from Section 3.1 performEnvironmentalChecks - Error Codes section updated from Section 3.2 InitializeSDK - Removed below Masterpass APIs and Business Objects from the document: ○ **APIs:** - getConsumerProfile - updateConsumerProfile - updateTermsAndConditions - Payment Card service APIs - Shipping Address service APIs - W2A checkout service APIs ○ **Business Objects:** - CheckoutShippingAddress - CheckoutPaymentCard - CheckoutInfo - PendingCheckoutResult - DeletePaymentCardResult - UpdatePaymentCardResult - AddPaymentCardsResult - CardBrandData - CardProductData - CardData - BillingAddress - UpdateCardDetails - CardDetails - UpdatePaymentCardInfo - PaymentCardInfo - DeleteShippingAddressResult - UpdateShippingAddressResult - ValidateCEPResult - AddShippingAddressesResult - ShippingAddressRequest - ShippingAddress - UserData - PaymentCard

Date	Version	Author	Comments
			- AuthorizeCheckoutData - CheckoutData - AddCheckoutShippingAddressesResult - UpdateCheckoutShippingAddressResult - MobileNumber - CheckoutNotification
29 May 2018	2.1.0	Mastercard	Below are the updated sections - • DEVICE_CPU_SUPPORTED check removed from performEnvironmentalChecks API • New input parameters added into initializeSDK API • Glossary • Description updated for handleNotification API. • startRecovery API specific changes - Parameter name renamed from gcmRegistrationID to mdesRnsRegistrationId & Parameter description updated. - Notes section updated. • createWallet API specific changes - Parameter name renamed from gcmRegistrationID to walletServerRnsRegistrationId & Parameter description updated. - Added new parameter cesRnsRegistrationId. - Notes section updated. • registerWithMdes API specific changes - Parameter name renamed from gcmRegistrationID to mdesRnsRegistrationId & Parameter description updated. - Notes section updated. • hasCWSPushContent API specific changes - Parameter type has been changed from Bundle to RemoteMessage. - Parameter name has been changed from data to remoteMessage. • hasMDESPushContent API specific changes - Parameter type has been changed from Bundle to RemoteMessage. - Parameter name has been changed from data to remoteMessage. • hasMDESPushContent API specific changes - Parameter type has been changed from Bundle to RemoteMessage. - Parameter name has been changed from data to remoteMessage. • getCWSPushContent API specific changes - Parameter type has been changed from Bundle to RemoteMessage. - Parameter name has been changed from data to remoteMessage. • Application wide configurations changes

Date	Version	Author	Comments
			- MDES_GCM_SenderId has been renamed to MDES_RNS_SenderId and description updated. - Wallet_GCM_SenderId has been renamed to Wallet_Server_RNS_SenderId and description updated. • CreateWalletOperationParams specific changes - gcmRegistratinId has been renamed to walletServerRnsRegistrationId & description updated. - Added new parameter “cesRnsRegistrationId”. • RegisterWithMdesOperationParams specific changes - gcmRegistratinId has been renamed to mdesRnsRegistrationId & description updated. Below is the removed section - • onRecoveryRequired callback event removed from Wallet Notifications. • Clean Up and Improved API doc • Removed below unused APIs - authenticate - validatePINFormat - changePIN - resetPIN - replenishPaymentToken • Sync Java Doc vs API doc • Below API has been added – - updateRnsRegistrationIds
04 September 2018	2.1.1	Mastercard	Below are the changes incorporated in Phase 2.1 R2: • Updated section Application wide configuration LogLevel with description related Logging priority and Default LogLevel. • Updated onTransactionError with below changes: – Removed INVALID_PAYMENT_TOKEN_STATUS and added NO_CARD_ACTIVE_FOR_PAYMENT. User will receive this error code when user tries to pay with a default card which was suspended earlier. • New VTS specific parameters added into TransactionDetails. • getTransactionHistory API response parameter description updated. • onTokenRefresh API is deprecated and MPA will receive callback in onNewToken.

Date	Version	Author	Comments
19 October 2018	2.1.2	Mastercard	Below are the changes incorporated in Phase 2.1 R3: • Marked “cesRnsRegistrationId” parameter as optional in createWallet and updateRnsRegistrationIds. • Added INVALID_CONFIGURATIONS error code in below APIs: – updateRnsRegistrationIds – startRecovery
20 November 2018	2.1.3	Mastercard	Below are the changes incorporated in Phase 2.1 R4: • Added enableFileBasedLogging section and updated Error Descriptions section with below new error code: – FILE_LOGGING_NOT_ALLOWED_IN_RELEASE_MODE – CONTEXT_NULL_WHILE_ENABLING_FILE_LOGGING • Added READ_PHONE_STATE_PERMISSION_REQUIRED error code in initiateTransaction API.
10 December 2018	2.1.4	Mastercard	**Note: This version has not been released to customer.** Below are the changes incorporated in Phase 2.1 R5: • Response Parameters description has been updated for the VISA payment token in getRemainingTransactionKeys API. • Updated enableFileBasedLogging section by adding LoggingProperties parameter. • Add new class LoggingProperties. • In digitize API, Updated Error Codes section. Added new error code as “VISA_PROVISION_FAILED”. • Added VISA_PROVISION_FAILED error code in Error Descriptions section. • Added Notes section in DigitizationFailedReason.
31 December 2018	2.1.5	Mastercard	Below are the changes incorporated Phase 2.1 R6: • In digitize API, Updated Error Codes section. Removed the “VISA_PROVISION_FAILED” error code and updated the description for the “PROVISION_FAILED” error code. • In digitize API, Updated the Notes section. • In digitizeByList API, updated the description section. • In getPaymentTokens API, Updated the Note section. • In DigitizationFailedReason, Updated the Note section. • Updated the Error Descriptions section. Removed the VISA_PROVISION_FAILED error code and Updated the description for PROVISION_FAILED error code.
16 April 2019	2.1.6	Mastercard	Below are the changes incorporated Phase 2.1 R8: • Added note in getRemainingTransactionKeys • Modified error codes in below APIs: – Deprecated UNKNOWN_PAYMENT_TOKEN, INVALID_PAN & PAN_INELIGIBLE error codes in getTermsAndCondition API. – Deprecated ACTIVATION_ERROR error code in requestActivationCode API.

Date	Version	Author	Comments
			– Deprecated ACTIVATION_ERROR & INVALID_WORKFLOW error codes in activate API. – Deprecated ACTIVATION_ERROR, INVALID_WORKFLOW & UNKNOWN_PAYMENT_TOKEN error codes in synchronize API. – Deprecated INVALID_WALLET_PIN error code in resumePaymentToken API. – Deprecated INVALID_WALLET_PIN error code in getTransactionHistory API. – Deprecated INVALID_WALLET_PIN error code in terminateWallet API. • Removed INVALID_CONFIGURATION and MDES_REGISTRATION_FAILED from updateRnsRegistrationIds. • Added APP_INSTANCE_TERMINATED and DEVICE_IS_ROOTED in following APIs – acceptSafetyNetFailedRisk – activate – synchronize – deletePaymentToken – suspendPaymentToken – resumePaymentToken – getTransactionHistory – startRecovery – updateRnsRegistrationIds – registerWithMdes – registerWithVts – digitize – digitizeByList – requestActivationCode
07 June 2019	2.1.7	Mastercard	Below are the changes incorporated in Phase 2.1 R9: • Removed READ_PHONE_STATE_PERMISSION_REQUIRED error code from below APIs as this error code has been deprecated: – checkDeviceEligibility – createWallet – getTermsAndCondition

Date	Version	Author	Comments
			– getTokenForOperation – registerWithMdes – registerWithVts – digitize – digitizeByList – requestActivationCode – activate – synchronize – deletePaymentToken – suspendPaymentToken – resumePaymentToken – initiateTransaction – getTransactionHistory – startRecovery – terminateWallet – updateRnsRegistrationIds – acceptSafetyNetFailedRisk • Removed READ_PHONE_STATE_PERMISSION_REQUIRED error code from Error Descriptions section as this error code has been deprecated. • Android Permissions excel file updated. • 'Removed force update RNS Ids when MPA upgrade scenario' from updateRnsRegistrationIds API.
13 April 2020	2.2.0	Mastercard	Below are the changes incorporated in Phase 2.2 • Added new attribute *Periodic_Sync_Delay_In_Months* in Application wide configurations. • Updated the description section for paymentSDK Input Parameter in initializeSDK API. • Removed the below APIs: – startRecovery – getRecoveryStatus • Removed RECOVERY_REQUIRED error code from the below APIs: – synchronize – registerWithMdes – registerWithVts

Date	Version	Author	Comments
			– digitize – digitizeByList – requestActivationCode – activate – terminateWallet – updateRnsRegistrationIds – getPaymentTokens – deletePaymentToken – suspendPaymentToken – resumePaymentToken – setDefaultPaymentToken – getDefaultPaymentToken – getRemainingTransactionKeys – initiateTransaction – getTransactionHistory • Added RESET_REQUIRED error code in below APIs: – synchronize – registerWithMdes – registerWithVts – digitize – digitizeByList – requestActivationCode – activate – terminateWallet – updateRnsRegistrationIds – getPaymentTokens – deletePaymentToken – suspendPaymentToken – resumePaymentToken – setDefaultPaymentToken – getDefaultPaymentToken – getRemainingTransactionKeys – initiateTransaction – getTransactionHistory • Description updated in reset API.

Date	Version	Author	Comments
			• Updated Error Descriptions by removing RECOVERY_REQUIRED and adding RESET_REQUIRED error codes. • Android Permissions excel file updated. Removed the GET_ACCOUNTS permission. • Data type is changed from Enum to String and description is updated for “transactionType” attribute in TransactionDetails section. • Added new attribute *transactionType* in TransactionInfo business object. • API Signature is changed and new parameter *emailAddress* added in the following APIs: – createWallet – digitize – digitizeByList • Updated the NOTE section regarding the SafetyNet check in APIs as follows: – initializeSDK – digitize – digitizeByList – Application Wide Configurations (Environment > Note) • Added new attribute *VTS_CvmPriorityConfiguration* in Application Wide Configurations section. • As part of DBL implementation changes, updated the below section: – Added new callback paymentTokenDigitized in Wallet Notifications - WalletNotificationListener

Date	Version	Author	Comments
			– digitizeByList - Updated the Description and Note section. – Added new attribute *errorReason* in DigitizationFailedReason.
8 July 2020	2.2.0.1	Mastercard	Description updated for Response Parameters Reference documents updated at References
21 July 2020	2.2.1	Mastercard	• Below API added – - digitize (Using Encrypted Card Information) - digitizeByList (Using Encrypted Card Information) • New Business Object added EncryptedCardInfo • Description & Error Codes section updated in digitizeByList API. Description & Note section updated in paymentTokenDigitized.

1.3 References

1	Mastercard Digital Enablement Service – MCBP Use Cases • Version 1.0.3
2	Mastercard Digital Enablement Service – API Specification • Version 1.1.3
3	flame_customizable_nfc_wallet_functional_specification_v2_2_0.pdf

1.4 Assumptions and Considerations

- All the long running operations would be an asynchronous call.
- All the local methods would be a synchronous call.
- Return values for asynchronous methods would be defined once LLD is done.
- Type of the input or output parameters may change for certain APIs. During the LLD, for certain APIs if the better API interface is identified then there may have change in input or output parameter. Note, the responsibility and operation would be kept same.
- Return of Error and its format (Return error via Exception or via return value) would be defined once LLD is done.
- Error messages returned are not Localized.

1.5 Glossary

The following abbreviations and acronyms are used in this document:

Abbreviation	Description
AES	Account Enablement System
API	Application Programming Interface
ATC	Application Transaction Counter
ID&V	Santander Authentication System

Abbreviation	Description
CMS -D	Credentials Management System Dedicated
CSR	Customer Service Representative
CVC	Card Verification Code
CVM	Cardholder Verification Method
DPAN	Device Primary Account Number
FCM	Firebase Cloud Messaging
CES	Card Enablement System
HCE	Host Card Emulation
MCBP	Mastercard Cloud Based Payment
MDES	Mastercard Digital Enablement Service
MNO	Mobile Network Operator
MPN	Mobile Phone Number
NFC	Near Field Communication
PAN	Primary Account Number
RNS	Remote Notification Service
TDS	Transaction Details Service
TUR	Token Unique Reference
MPA	Mobile Payment Application (developer)

1.6 Definition

This section explains a number of key terms and concepts used in this document:

Term	Meaning
Account PAN	The cardholder's real primary account number; PAN used by the issuer to fund the transaction.
Authorization Token	Token generated by the Santander ID&V system and returned to the WL-MPA after a successful user authentication, see [1]
Card Availability	A check to determine whether a card is generally within an available range for digitization.
Card Assets	They are the card images and the product names to be displayed in the Mobile Wallet.
Card Eligibility	A check to determine whether a specific card is eligible for digitization.
Card State – Active	This state indicates that a payment card is successfully approved and provisioned to user. The user will be able to see colored payment card and will be able to use it for payment

Term	Meaning
Card State – Deleted/Removed	This state indicates that user will no longer be able to use the card. The card deletion/removal can happen in the following ways: • By User: The user wants to discontinue using a payment card and opts to remove it from the wallet. The Wallet Server will delete the card. The card will disappear from the Mobile Wallet. • By CSR: The CSR deletes the card via MDES portal. The Wallet Server will delete the card and notify the Mobile Wallet. The card will disappear from the Mobile Wallet. • By Wallet Server: This is triggered by execution of the "Terminate Wallet – By Wallet Server" use case wherein a request to register Wallet is received from a device for which an "Active" Wallet exists on the Wallet Server. The Wallet Server terminates the existing record and deletes the associated cards and proceeds with the new registration request.
Card State – Suspended	This state indicates that an authorized administrator (on behalf of bank, issuer) has temporarily suspended the payment card due to certain business or technical reasons. Once the cause is resolved, the card can be resumed. There shall be some appropriate visual indicator to identify the suspended card.
Contactless	A wave-and-pay system that employs RFID technology, and allows shoppers to pay by touching their device against an electronic reader
Credentials Management System	The Credentials Management System (CMS) is responsible for provisioning data from the Account Enablement System, creating Transaction Credentials and delivers them to the Mobile Payment App on the Device. The Credentials Management System is split architecturally into two components: <u>Core</u> – It holds the master keys for the Token, from which Session Keys are derived. <u>Dedicated</u> – which receives the Session Keys from Core, combining them with the Mobile PIN (or other CVM) and the device profile to transform them into Transaction Credentials for use by the Mobile Payment App.
Credentials Management System	Accepts provisioning data from the Account Enablement System, creating Transaction Credentials and delivers them to the Mobile Payment App on the Device.
Digitization	Process of personalization and provisioning of account credentials, cryptographic keys (EMV/Chip), and associated data, into digital devices
Google Cloud Messaging	Google Cloud Messaging is a service that allows a server to send data to an Android-powered device, and also to receive messages from devices on the same connection. The Google Cloud Messaging service handles all aspects of queuing of

Term	Meaning
	messages and delivery to the target Android application running on the target device.
Host Card Emulation	Host Card Emulation allows the Device to act as a chip card without requiring a chip based Secure Element. One option to implement HCE is for a software-based application in the Device host to interact with the NFC Controller. The Host Card Emulation functionality may be called by different names depending on the Device Operating System. For simplicity, this document refers to the functionality as Host Card Emulation, and does not imply a specific implementation or Device Operating System
Identification and Verification (ID&V)	Process of identifying and verifying a cardholder prior to initiating digitization
Issuer	The financial institution that issues payment cards and holds the account or credit line behind the card
Mastercard Digital Enablement Service	Mastercard digitization and tokenization platform.
Mobile Payment Application (MPA) or Mobile Wallet	A mobile payment Application providing User experience and User interface for Contactless transaction. A Mobile Payment Application can be loaded with more than one Digitized Card.
Device	The Device is a smartphone owned by the User. The Device has the ability to download applications from a standard Mobile Operating System Application Store, such as Google Play
Device Fingerprint	The device Fingerprint is defined as a list of information collected about the Device and the Mobile Payment Application running on that Device. It is used for the purpose of binding information with the Device and the instance of the Mobile Payment Application actually used by the user.
Mobile PIN	A personal value shared between User and issuer (not an offline PIN, or mPIN) for a Payment Card. Can be the same as the cardholder’s online PIN or any other value. The Mobile PIN is always required for both low and high value transactions initiated with the Mastercard Cloud-Based Payments token. The Mobile PIN is entered by the cardholder into the device, but the Mastercard Cloud-Based Payments Mobile Wallet does not validate the Mobile PIN offline; it is implicitly validated online as part of the cryptogram validation. Incorrect Mobile PIN entry will result in a malformed cryptogram being generated. The Mobile Wallet application does not have any support of offline validation of the Mobile PIN value. The mobile application has no other means than doing an online transaction to know whether the Mobile PIN value was correct

Term	Meaning
	or not. That way the issuer (Mastercard) controls the Mobile PIN validation process and can apply some fraud detection techniques (using for example the concept of a Mobile PIN Try Counter). This applicable for Mastercard cards only for the project.
Near Field Communication (NFC)	Near field communication (NFC) is a set of standards for smartphones and similar devices to establish radio communication with each other by touching them together or bringing them into proximity, usually no more than a few inches
Play Store	Play store is a type of digital distribution platform for mobile apps.
RNS	It stands for Remote Notification Service. It is a platform-specific service used to send data from a server to a Device. Examples of Remote Notification Services include Google Cloud Messaging, Apple Push Notification Service, or Microsoft Push Notification Service.
Suspend	Token has temporarily been suspended and cannot make a payment transaction.
Token	Surrogate value for an Account PAN that is a 13 to 19-digit numeric value that is associated to a consumer’s card.
Token Credential	The credential (for example, Card Profile) for the Token as generated by the MDES.
PaymentToken	Same as Token Credential. Entity/Object named for Token Credential in CW-SDK API.
Token Unique Reference (TUR)	Reference returned by MDES to the Wallet Server / Corporate Wallet SDK uniquely identifying the token of a card that has been digitized. It is needed by the Corporate Wallet SDK to fetch the transaction details or the transaction history from TDS.
Token Vault	A repository that maintains the established Token to Account PAN mapping.
Tokenization	Tokenization is the process of replacing the Cardholder Primary Account Number (PAN) with a surrogate value (that is, a Tokenized PAN) that is used in place of the PAN in payment transactions.
Tokenization Authentication Value (TAV)	A cryptographic authentication value that is generated by the Issuer and verified by the Mastercard Enablement System to authorize a digitization request.
Transaction Details Service (TDS)	Service provided by MDES in order to retrieve the transaction details after a transaction has been done or to receive the transaction history related to a specific token (i.e. card) digitized in the MPA.
User	The labels Consumer / User / Cardholder refer to the same entity, a User of the NFC Wallet.

Term	Meaning
Wallet PIN	Wallet PIN is a personal value shared between User and Wallet server for Wallet User authentication.
Wallet Server	Wallet Server is the entity that manages the Wallet lifecycle, provides services for managing front end experience. Wallet server also provides integration with MDES Digitization APIs.
Wallet State – Active	The Wallet state transitions to this state from “In-active” state to “Active” when response to registration is prepared by Wallet Server.
Wallet State – In-active	The Wallet state is “In-active” when the PIN set-up is done at Wallet Server and response to registration is not prepared by Wallet Server.
Wallet state - Stateless	The Wallet State is stateless when the wallet has been downloaded and initiated and PIN set up pending.
Wallet State = Terminated/Deactivated – By Server	This is an end-state, in other words, wallet cannot be recovered from this state and user can no longer use the wallet. The Wallet Termination shall deletion/removal can happen in the following ways: • By User: The user has an option to terminate/de-activate the Wallet from the Mobile Wallet. When the user exercises this option, the Wallet Server shall terminate the Wallet record and all the data from the Mobile Wallet shall be removed. The user can once again set up a Wallet by undergoing Wallet set up process without having to download and install the application. • By Wallet Server – The Wallet Server will terminate Wallet record if it a request to register Wallet is received from a device for which an “Active” Wallet exits on the Wallet Server. The Wallet Server terminates the existing record and deletes the associated cards and proceeds with the new registration request.

2 Solution Overview

The Mobile Payment Application (MPA) is the consumer wallet app that supports In-store Payments through NFC HCE contactless tap and pay.

The MPA will be built by Santander regional banks using the Corporate Wallet SDK. The first version of the Corporate Wallet SDK will only support the In-Store payments (#1 above) but will subsequently be enhanced to include the other forms of wallet payment.

The Corporate Wallet SDK will be available for Android ONLY. It encompasses a number of sub-components (as black boxes) to offer a unified API for all wallet related functionalities. All server interactions related to the wallet services will also be handled internally by the APIs.

The diagram below provides an SDK Centric view of the proposed solution.

Figure 1. Solution Architecture

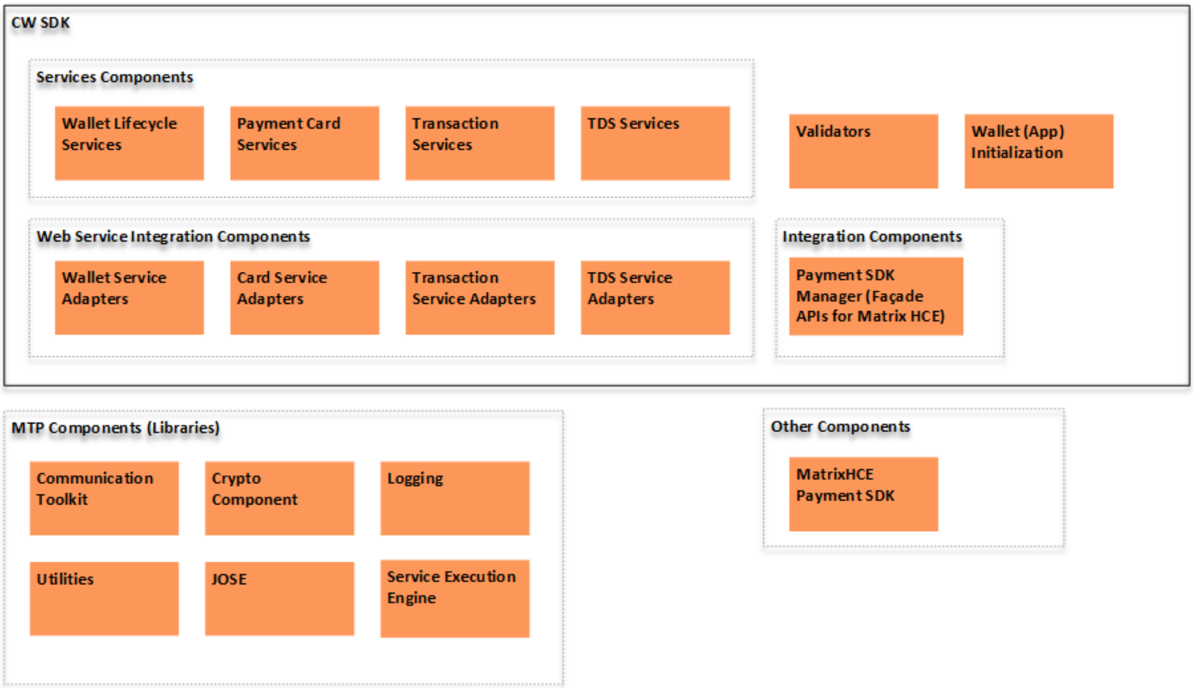
Sr. No.	Component	Description
1	Santander MPA	This is a consumer reference application that is developed using the Corporate Wallet SDK to provide wallet functionality to the consumer. Integration is performed with 1. Wallet Server: for Wallet services. 2. MDES CMS-D/ Mobile Payment APIs: for card and token provisioning 3. MDES Transaction Detail Service: for transaction details and history services 4. VTS for token provisioning, transaction details and history services The Santander-MPA is composed of: - WL-MPA: mainly responsible for the UI, T&C management, Santander ID&V user authentication, and user card selection of to be digitized cards - Corporate Wallet SDK: It provides all the Wallet functionalities. It encapsulates other components that includes the Inside Secure URPay Payment SDK.
2	Wallet Server	The Wallet Server will support the Wallet lifecycle management and will integrate with MDES and VTS Digitization APIs to enable payment card digitization.
3	MDES	Tokenization and Digitization infrastructure provided by Mastercard.
4	VTS	Tokenization and Digitization infrastructure provided by Visa.
5	Santander Systems	It is the Santander systems with which the Wallet Server and/or Mobile Wallet shall communicate for specific tasks: • Notify token status update to Card Enabler System (CES)

Sr. No.	Component	Description
		• Authenticate user using ID&V • SPS – To validate authorization token
6	FCM	It is Firebase Cloud Messaging services for sending push notifications to MPA.

2.1 Component View

This section discusses the overall structure for CW-SDK in terms of the grouping of components into separate logical as well as physical layers.

Figure 2. CW-SDK Component View



As shown in the above figure CW-SDK for Android Mobile application consists of the following components. It also shows all the other components it uses to perform its functions.

Component	Description
CW-SDK	It provides various Wallet services through a façade API which can be used by client components such as WL-MPA to achieve various Wallet business use cases. Refer below section for list of services provided by this component.
Inside Secure URPay SDK	Mobile Payment SDK for management of Mastercard and Visa branded Tokens and payment

Component	Description
Service Engine toolkit	Provides abstractions for business use cases or service implementations. Hides the complexity of service security, reliability, transactions and traceability. This is an MTP Component and would be used as a Library.
Communication Toolkit	This component hides the complexity of exchanging the data between Client and Server by providing high-level communication APIs. It also takes care of securing the data via JOSE during the communicating data between client and server. This is an MTP Component and would be used as a Library.
Logging toolkit	Provides simple but powerful logging APIs for mobile client. Facilitates capturing of logs on various logging channels like Console, File, DB, Remote logging. Provides an option to turn ON/OFF via configurations. This is an MTP Component and would be used as a Library.
Utilities	The mobile client utilities toolkit is envisaged as a single umbrella to provide a bunch of utilities that help developers to understand various characteristics of a mobile application and its execution environment. It helps in risk-based decision making while implementing business solutions. It also provides various APIs to implement other peripheral use cases such as, auto-log off in case user is not interacting with mobile. App for certain time period. This is an MTP Component and would be used as a Library.

CW-SDK Library is composed of below logical components and provides below Services

Component	Description
Wallet Lifecycle Services	Business component encapsulating wallet lifecycle services and exposing simple as well as composite APIs for wallet services. Following key APIS are provided: • Check device eligibility • Create Wallet • Digitize and other APIs. Please refer API specification for further details.
Transaction Services	Provides following services • Payments/Refunds • GetTransactionHistory and GetReciept • Notification handling during Payments and other services. Please refer API specification for further details.

Component	Description
TDS Services	Provides following services - Registration with TDS - Fetching Transaction History.

3 Wallet SDK Services

3.1 performEnvironmentalChecks (New Static API)

This API is used to perform various environmental checks of device and application to ensure if the environment is eligible and secure.

WL-MPA should always use this API before **CW-SDK** initialization and then check for the different environmental support and if support is as per the requirement of MPA then only initialize the else MPA can show relevant error screen or message to the user.

CW-SDK performs following checks and operations when this API is used and returns the relevant result:

- Checks if the application is installed from authorized source check.
- Rooted device check.
- NFC is supported and enabled on the device.
- Secure unlock mechanism on device is enabled or not (device is secured with a PIN, pattern or password and other lock mechanism)

This API returns a list of various environmental checks performed along with its results. See Environment Checks for more information.

3.1.1 Input Parameters

Parameter	Type	Description
context	Context	Android application context.
checks	List having Environment Check constant values	List of checks to be performed. If no checks are passed in parameter list, it would perform all the checks and result of those are returned.

3.1.2 Response Parameters

Parameter	Type	Description
environmentCheckRelsults	Map	A Map of the operation/checks performed and its result. Map holds key-value pair where key is a constant mentioned in Environment Check. The value will be **TRUE** if the operation is passed, otherwise the value will be FALSE. - Device Rooted - NFC Supported - Authorize Source installation - NFC Enabled - Secure Unlock Mechanism Enabled

Notes:

- **This API would perform all the local device and application health checks and returns it results.**
- **This method will throw runtime exception (IllegalArgumentException) with message as “INVALID_INPUT: Context cannot be null” If the android context passed is null. Android context should not be null.**

Use-Case:

- Application Launch.
- Tap & Pay.

3.2 initializeSDK

This method is used to initialize and setup the CW-SDK with configurations passed to it. This API returns an initialized CW-SDK instance object that can be used to call all the CW-SDK APIs/methods.

WL-MPA should always call performEnvironmentalChecks API before *CW-SDK* initialization and then check for the different environmental support and if support is as per the requirement of MPA then only initialize the *CW-SDK* else MPA can show relevant error screen or message to the user.

WL-MPA must have to initialize the Payment SDK before *CW-SDK* initialization and then pass the same object to initializeSDK API. Also, WL-MPA must have to create and pass instance of ICommsRequestProcessor during initializeSDK*.*

WL-MPA should pass the required configuration for initializing and setting up the CW-SDK. These configuration are instance specific which can't be changed across the life of the application.

CW-SDK would also use few of the application wide configurations that are defined in configuration file and bundled with application during the build time. Refer Application wide configuration options for more information.

CW-SDK would cache these configurations and use it in various business functions.

This method call is mandatory and should be called by WL-MPA for proper functioning of all the future calls to CW-SDK methods.

3.2.1 Input Parameters

Parameter	Type	Description
context	Context	Android application context
configuration	WalletSdkConfiguration	Various configurations to initialize the CW-SDK
commsRequestProcessor	ICommsRequestProcessor	Communication processor for wallet server request and response. For example, MPA can plug default *CommsRequestProcessor* during *CW-SDK* initialization.
paymentSDK	PaymentSDK	Initialized Payment SDK object which will be used for payment transaction. For example, MPA can plug default *URPaySDKImpl* during *CW-SDK* initialization. **Note:**

Parameter	Type	Description
		In case if this value is Null, CW-SDK will throw the runtime exception (IllegalArgumentException). URPay SDK must be initialized successfully first in order to call this API. Please refer "CWSDK Integration Guide" to learn more about initialization of URPay.

3.2.2 Response Parameters

Parameter	Type	Description
walletSdk	WalletSdk	Returns CW-SDK instance object.

3.2.3 Error Codes

Error Code	Description
INITIALIZATION_FAILED	If Initialization fails due to failure of internal system. (For example, ServiceEngine initialization failed).
INVALID_CONFIGURATIONS	If mandatory configurations are missing or invalid during CW-SDK initialization.

Notes:

- **This method only initializes the CW-SDK and other dependent components for proper functioning of other methods.**
- **This method does not perform any other environmental checks or operations. These checks can be performed using performEnvironmentalChecks method on the object returned in this method.**
- **This method will throw runtime exception (IllegalArgumentException) with message as "INVALID_CONFIGURATION + Description of the invalid configuration" If the mandatory configurations are not passed or any invalid configurations are passed.**
- **CW-SDK performs SafetyNet check asynchronously in this API call when the environment is other than DEV and WalletStatus is ACTIVE.**
- **Refer "CWSDK Integration Guide" to learn more about initialization of URPay SDK and CW-SDK.**

Use-Case:

1. Application Launch.
2. Application Launch during Tap & Pay.

3.3 getInstance

This method is used to retrieve the singleton instance of CW-SDK object that was created and initialized during initializeSDK method.

This method would return a CW-SDK instance or throw an exception if it is called before initializing the CW-SDK.

3.3.1 Input Parameters

Parameter	Type	Description
Not Input Parameters		

3.3.2 Response Parameters

Parameter	Type	Description
walletSdk	WalletSdk	Returns CW-SDK instance object.

3.3.3 Error Codes

Error Code	Description
SDK_NOT_INITIALIZED	If CW-SDK is not initialized. **Note: CW-SDK can be initialized using initializeSDK method.**

Notes:

- **To call various APIs/methods of CW-SDK an instance object of CW-SDK is required which can be retrieved using this method.**
- **This method does not perform any other operations.**

Use-Case:

- To call all the APIs/methods of CW-SDK

3.4 checkDeviceEligibility

This method is used to check device and application eligibility before creating the Wallet and making digitization request. It is used to check device eligibility including whether or not the device is a Mastercard type-approved device, hardware and software compatibility and any applicable Issuer policies related to the device.

This method will return TRUE if device is approved as per all eligibility criteria.

This method would make a remote call to Wallet Server for eligibility check. Internally in CW-SDK, it captures few of the Device and Application attributes and would send it to Wallet Server as part of eligibility request. WS would perform the relevant eligibility checks and responds back the result to CW-SDK.

3.4.1 Input Parameters

Parameter	Type	Description
Not Input Parameters		

3.4.2 Response Parameters

Parameter	Type	Description
result	Boolean	Result of the check eligibility check.

3.4.3 Error Codes

Error Code	Description
NO_DATA_CONNECTIVITY	No Data connectivity available.
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
COMMUNICATION_ERROR	Error while communicating with remote server **Note: May get various system error response from server. All would be collated under this error code**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure

Note:

This method can be called by MPA based on the requirement.

Use-Case:

- On Application Launch only if wallet status is INACTIVE.

3.5 synchronize

This method is used to synchronize the state and data between CW-SDK and Wallet Server.

This method would make a remote call to Wallet Server by passing the last updated timestamp (CW-SDK itself internally manages time stamp. No need to manage by WL-MPA). Wallet Server would identify last timestamp and return all the changes done on the wallet account post last updated timestamp to CW-SDK.

Post successful synchronization, CW-SDK would notify MPA with the results of synchronization and then MPA can call various other APIs to get the updated data from CW-SDK.

Information such as PaymentToken information and other wallet state information such as availability of new MPA version would be synchronized between CW-SDK and Wallet Server.

This method will also do the reliability handling for any unfinished past operations and make the operation and its relevant data in consistent state. For example, this method will synchronize card digitization status in case the digitization of any card failed due to network failure or any other unexpected failure.

3.5.1 Input Parameters

Parameter	Type	Description
Not Input Parameters		

3.5.2 Response Parameters

Parameter	Type	Description
syncResult	SyncResult	Result of Synchronization call.

3.5.3 Error Codes

Error Code	Description
NO_DATA_CONNECTIVITY	No Data connectivity available.
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
COMMUNICATION_ERROR	Error while communicating with remote server **Note: May get various system error response from server. All would be collated under this error code**
~~UNKNOWN_PAYMENT_TOKEN~~	~~PaymentToken is unknown.~~ **Note: This error code is deprecated.**
~~INVALID_WORKFLOW~~	~~There is any error in workflow.~~ ~~For example, Card activation is called before requesting for activation code, SUSPEND or RESUME called on INACTIVE Token and other workflows.~~ **Note: This error code is deprecated.**
~~ACTIVATION_ERROR~~	~~It could be due to other reason like session expired.~~ **Note: This error code is deprecated.**
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device.
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Notes:

- **This method should be called by WL-MPA post launch of an application to synchronize the application state and data between CW-SDK and WS.**
- **Optionally, it can further be called by WL-MPA at later stages of application life based on the business requirements. For performance optimization, synchronization is not automatically triggered for all calls to the API, but is let to WL-MPA decision.**

3.6 handleNotification

This method is used to forward the push notification received by WL-MPA to CW-SDK.

WL-MPA will do the RNS registration and hence all push messages would be received by WL-MPA. It is WL-MPAs responsibility to forward all CW-SDK specific messages to CW-SDK via this API call based on Sender-ID. Messages from Wallet Server and MDES (CMS-D) shall always be passed to CW-SDK.

Once CW-SDK receives the push message, the CW-SDK will parse the message and perform various business operation based on the message.

3.6.1 Input Parameters

Parameter	Type	Description
from	String	Sender information as received from Android API
payload	String	Bundle information as received from Android API

3.6.2 Response Parameters

Parameter	Type	Description
No response parameters		

3.6.3 Error Codes

Error Code	Description
INVALID_INPUT	Required input arguments are invalid.
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure

Note:

If MPA fails to forward the CW-SDK specific push messages to CW-SDK, the behavior and state of CW-SDK (and hence MPA) would be unknown.

3.7 reset

This method is used to wipe out the data stored in CW-SDK. A call to this method will clear all the information like Wallet instance, Payment Cards etc. from CW-SDK and will bring it back to its initial state.

While this method wipes out CW-SDK data, it will not wipe out the configurations provided by MPA during initializeSDK API call. Meaning, re-initialization of CW-SDK (by calling initializeSDK API) is not required to perform other operations after the reset operation. MPA must call this API when receives RESET_REQUIRED error from CW-SDK APIs.

Note:

Because reset API wipes out all the information from CW-SDK, it is strongly recommended to exercise caution while using this API.

3.7.1 Error Codes

Error Code	Description
RESET_FAILED	CW-SDK reset failed. Please retry again.

4 Wallet Services

4.1 createWallet

This API is used to create Wallet instance on Wallet Server for a given user and on a given device.

Actual user registration is a responsibility of the WL-MPA and would be done separately along with T&C acceptance with the bank's back-end systems. This method is only used to initialize the wallet server with a new wallet record and associate it to the device.

Wallet Server will authenticate and authorize a user via AuthToken value provided by the WL-MPA using the bank deployed ID&V systems. In the case of invalid or expired AuthToken, Wallet Server will provide a proper exception via CW-SDK.

This method also requires PIN to be passed which would be used to setup as a Wallet PIN. In the scenario, where the PIN is not required and hence not passed by MPA, the PIN would be generated internally either on CW-SDK or Wallet Server and same would be used. For security reasons, received PIN would be transformed by salting and hashing before sending it to the Wallet Server. Further, for remote notifications handling, this method requires RNSID from WL-MPA which would then be passed to Wallet Server as a part of wallet creation request.

This method would capture few of the Device and Application attributes and would send it to Wallet Server as part of registration request. These attributes are required for verification during all business communications between CW-SDK and Wallet Server to happen in future.

It performs following operations:

- Requests the Wallet Server to create a Wallet instance for given user for this device.
- On successful response, changes the state of the application to WALLET_ACTIVE.

WL-MPA can also retrieve Wallet instance information via getWalletInfo API or can terminate via terminateWallet API.

4.1.1 Input Parameters

Parameter	Type	Description
authToken	AuthToken	Authorization token received from Santander ID&V after successful authentication. This object will be composed of identity token and data token
walletServerRnsRegistrationId	String	RNS Identifier registered by the WL-MPA for Wallet Server.
cesRnsRegistrationId	String	RNS Identifier registered by the WL-MPA for CES. Conditional – Required if MPA country has CES system
walletPIN	Byte [ ]	Wallet PIN. Conditional – Required if PIN is managed by WL-MPA
emailAddress	String	Primary E-mail account address configured in mobile device.

Parameter	Type	Description
		NOTE: • Before calling this API, every-time MPA should programmatically fetch the primary email account address configured in the mobile device. To check how to obtain the configured email account address from a mobile device, refer *Wallet Management Overview → Create Wallet → NOTE* section in the Integration Guide. • This is an optional parameter and CW-SDK doesn't validate the emailAddress. • This emailAddress is used by the Wallet Server to generate the digitization recommendation. Then Wallet Server send this recommendation to the ISSUER via MDES. ISSUER provides the digitization decision based on this recommendation. Check *decision* in PaymentToken for more details.

4.1.2 Response Parameters

Parameter	Type	Description
walletStatus	Enum	Wallet status after the operation is performed. It would be either ACTIVE, INACTIVE, or other wallet status.

4.1.3 Error Codes

Error Code	Description
NO_DATA_CONNECTIVITY	No data connectivity available.
DEVICE_NOT_SUPPORTED	Device not supported.
OS_NOT_SUPPORTED	OS is not supported.
INVALID_INPUT	Required input arguments are invalid.
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code.**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
AUTH_TOKEN_AUTHORIZATION_FAILED	ID&V token authorization failed.

Notes:

- **Before calling the API, Authorization with ID&V should be done.**
- **To check how to obtain the registration ID, please refer Firebase Cloud Messaging Configuration section in the integration guide.**

- **If wallet creation is being done again from the same device. For example, after reset API call, WS would remove all the PaymentTokens for this device prior to creating wallet again. This means that WS would terminate the earlier wallet and then create new wallet record.**

Reference use cases

- Register Wallet
- Activate Wallet

4.2 registerWithMdes

This method is used to register wallet with (MasterCard Digital Enablement Service) MDES system. This method allows WL-MPA to enroll a device and application with MDES to provision tokens on the device. WL-MPA must call this API before performing digitization of Mastercard cards.

This method would make a request to Wallet Server, which in turn would connect with MDES system to register this application instance. On successful response from Wallet Server, CW-SDK would do the required initialization to store digitized payment tokens.

4.2.1 Input Parameters

Parameter	Type	Description
authToken	AuthToken	Authorization token received from ID&V after successful authentication.
mdesRnsRegistrationId	String	RNS Identifier registered by the WL-MPA for MDES.

4.2.2 Response Parameters

Parameter	Type	Description
No Parameters Required.		

4.2.3 Error Codes

Error Code	Description
AUTH_TOKEN_AUTHORIZATION_FAILED	ID&V token authorization failed.
MDES_REGISTRATION_FAILED	Registration with MDES fails.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
WALLET_ALREADY_REGISTERED	Wallet is already registered with MDES.
NO_DATA_CONNECTIVITY	No data connectivity available.
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
INVALID_INPUT	Required input arguments are invalid.

Error Code	Description
COMMUNICATION_ERROR	Error while communicating with remote server. **Note: May get various system error response from server. All would be collated under this error code**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
INVALID_CONFIGURATIONS	If the mandatory configurations are not passed or any invalid configurations are passed for MDES system.
AUTH_TOKEN_CREDENTIAL_MISMATCH	ID&V credentials passed during registerWithMdes call is not same as ID&V credentials used during wallet creation.
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error occurs in case when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Notes:

- **Prior to calling this API, WL-MPA must have previously created the wallet using *createWallet* API.**
- **Before calling the API, Authorization with ID&V should be done.**
- **To check how to obtain the registration ID, please refer Firebase Cloud Messaging Configuration section in the integration guide.**

Reference use cases:

1. Register Wallet
2. Activate Wallet

4.3 registerWithVts

This method is used to register wallet with Visa Token Service (VTS) system. This method allows WL-MPA to enrol a device and application with VTS to provision tokens on the device. WL-MPA must call this API before performing digitization of Visa cards.

This method would make a request to Wallet Server, which in turn would connect with VTS system to register this application instance. On successful response from Wallet Server, CW-SDK would do the required initialization to store digitized payment tokens.

4.3.1 Input Parameters

Parameter	Type	Description
authToken	AuthToken	Authorization token received from ID&V after successful authentication.

4.3.2 Response Parameters

Parameter	Type	Description
No Parameters Required.		

4.3.3 Error Codes

Error Code	Description
AUTH_TOKEN_AUTHORIZATION_FAILED	ID&V token authorization failed.
VTS_REGISTRATION_FAILED	Registration with VTS fails.
WALLET_ALREADY_REGISTERED	Wallet is already registered with VTS.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
NO_DATA_CONNECTIVITY	No data connectivity available.
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
INVALID_INPUT	Required input arguments are invalid.
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code.**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
INVALID_CONFIGURATIONS	If the mandatory configurations are not passed or any invalid configurations are passed.
AUTH_TOKEN_CREDENTIAL_MISMATCH	ID&V credentials passed during registerWithVts call is not same as ID&V credentials used during wallet creation.
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device

Error Code	Description
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Notes:

- **Prior to calling this API, WL-MPA must have previously created the wallet using *createWallet* API.**
- **Before calling the API, Authorization with ID&V should be done.**

Reference use cases:

1. Register Wallet
2. Activate Wallet

4.4 digitize

This method is used to perform digitization of card. It returns the processed PaymentToken having digitization status along with display attributes for that user.

This method requires AuthToken and CardInfo that needs to be digitized. It would make a request to Wallet Server for digitization of card, which in turn would call the dependent systems to get it digitized.

In response from WS to CW-SDK, for Mastercard cards, Card (PaymentToken) information along with other card art assets would be received. And for Visa Cards, relevant information for digitization would be received which would be handed over to Visa SDK for personalization.

Remotely during the card digitization process for Mastercard cards, CMS-D (part of MDES system) would raise a push notification for provisioning of PaymentTokens on this device, which should be received by CW-SDK. On receiving the push notification, CW-SDK would proceed with provisioning of the PaymentToken on the device.

In case of Yellow path (Additional authentication required), this method would return PaymentToken along with list of available authentication methods. In case of authentication method is ‘activation using activation code’, WL-MPA would have to request for activation code using *requestActivationCode* method and call the ‘activate’ method to proceed for digitization.

Once the PaymentToken is provisioned on device and is ACTIVE, CW-SDK would perform replenishment of transaction credentials. In case of Mastercard card, it would also make a call to WS for registration with TDS.

Note:

A Card (Account PAN) may only be provisioned once to a given application unless it has previously been removed. If a PaymentToken (active, suspended, or inactive pending activation) for that Account PAN is found for the given App, the digitization will be declined by this API.

4.4.1 Input Parameters

Parameter	Type	Description
authToken	AuthToken	Authorization token received from Santander ID&V after successful authentication. This object will be composed of identity token and data token
termsAndConditionsAssetId	String	Information related to Terms & Conditions acceptance. This is optional parameter.
cardInfo	CardInfo	Information of card to be digitized Refer CardInfo section in Appendix for details of CardInfo object.
decisioningData	DecisioningData	Contains data relevant for digitization decision. This is required to be passed to MDES & VTS in digitize request
emailAddress	String	Primary E-mail account address configured in mobile device. **NOTE:** • Before calling this API, every-time MPA should programmatically fetch the primary email account address configured in the mobile device. To check how to obtain the configured email account address from a mobile device, refer *Wallet Management Overview → Create Wallet → NOTE* section in the Integration Guide. • This is an Optional Parameter and CW-SDK doesn't validate the emailAddress. • This emailAddress is used by the Wallet Server to generate the digitization recommendation. Then Wallet Server send this recommendation to the ISSUER via MDES. ISSUER provides the digitization decision based on this recommendation. • This emailAddress should be same as used during the Create Wallet API call. In case, if the different emailAddress is used or primary e-mail account is changed after create wallet, then ISSUER may (or may not) give the REQUIRE_ADDITIONAL_AUTHENTICATION or DECLINED decision for the digitized card (depending upon the

Parameter	Type	Description
		business). Check decision in PaymentToken for more details.

4.4.2 Response Parameters

Parameter	Type	Description
paymentToken	PaymentToken	PaymentToken (Digitized Card) information would be returned. Refer PaymentToken section in Appendix for details of PaymentToken object.

4.4.3 Error Codes

Error Code	Description
AUTH_TOKEN_AUTHORIZATION_FAILED	ID&V token authorization failed.
WALLET_NOT_REGISTERED_WITH_TSP	Wallet is not registered with MDES/VTS.
TNC_REQUIRED	Terms and Conditions is required to digitize the card.
TNC_ACCEPTANCE_FAILED	Terms and Conditions acceptance failed on WS or MDES/VTS.
MISSING_EXPIRY_DATE	The expiry date is required for this card product but was missing.
NO_DATA_CONNECTIVITY	No data connectivity available.
PAN_INELIGIBLE_FOR_DEVICE	The PAN is not allowed to be provisioned to the device because of Issuer rules.
CARD_ALREADY_DIGITIZED	Card was already digitized previously.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
PAN_PROVISIONING_COUNT_EXCEEDED	The PAN has already been provisioned to the maximum number of devices.
DEVICE_INELIGIBLE	The device is not supported for use with MDES/VTS.
INVALID_PAN	The PAN format is not valid, or other data associated with the PAN was incorrect or entered incorrectly.
PAN_INELIGIBLE	The PAN is not in an approved account range for MDES/VTS.
INVALID_INPUT	Required input arguments are invalid.
COMMUNICATION_ERROR	Error while communicating with remote server.

Error Code	Description
	Note: **May get various system error response from server. All would be collated under this error code.**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
PROVISION_FAILED	The provisioning of PAN to the device has failed. Please delete this card. **Note:** **When this exception received, Its responsibility of MPA to delete that payment token.**
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note:** **This exception will only occur in the case of Visa.**
AUTH_TOKEN_CREDENTIAL_MISMATCH	ID&V credentials passed during digitize call is not same as ID&V credentials used during wallet creation.
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Notes:

- **Prior to calling this API, WL-MPA must have previously created the wallet using createWallet method and registration with associated TSP should be done using registerWith*MDES* or registerWith*VTS* method.**
- **In scenario when Terms and Conditions is required for digitization of Card then this method would return with TNC_REQUIRED error. Terms & Conditions can be fetched from server using getTermsAndConditions API/method. On confirmation of Terms & Conditions, WL-MPA would have to call this method again passing Terms & Conditions acceptance information to it.**
- **If this API throws "PROVISION_FAILED" exception, then MPA should delete that PaymentToken by calling deletePaymentToken API.**

- **CW-SDK performs SafetyNet check asynchronously in this API call when the environment is other than DEV and WalletStatus is ACTIVE.**

Reference use cases:

1. Digitize use case
2. Add Card use case

4.5 digitize (Using Encrypted Card Information)

This method works similar as the digitize API and used to perform the digitization of card while using the **Encrypted Card Information**. The WL-MPA would receive the **Encrypted Card Information** from the Santander Server**.** This API returns the processed PaymentToken having digitization status along with display attributes for that user.

This method requires AuthToken and EncryptedCardInfo that needs to be digitized. It would make a request to Wallet Server for digitization of card, which in turn would call the dependent systems to get it digitized.

In response from WS to CW-SDK, for Mastercard cards, Card (PaymentToken) information along with other card art assets would be received. And for Visa Cards, relevant information for digitization would be received which would be handed over to Visa SDK for personalization.

Remotely during the card digitization process for Mastercard cards, CMS-D (part of MDES system) would raise a push notification for provisioning of PaymentTokens on this device, which should be received by CW-SDK. On receiving the push notification, CW-SDK would proceed with provisioning of the PaymentToken on the device.

Decision value as REQUIRE_ADDITIONAL_AUTHENTICATION indicates Yellow path for which WL-MPA should follow the same process mentioned in digitize API call.

Once the PaymentToken is provisioned on device and is ACTIVE, CW-SDK would perform replenishment of transaction credentials. In case of Mastercard card, it would also make a call to WS for registration with TDS.

Note:

A Card (Account PAN) may only be provisioned once to a given application unless it has previously been removed. If a PaymentToken (active, suspended, or inactive pending activation) for that Account PAN is found for the given App, the digitization will be declined by this API.

4.5.1 Input Parameters

Parameter	Type	Description
authToken	AuthToken	Authorization token received from Santander ID&V after successful authentication. This object will be composed of identity token and data token
encryptedCardInfo	EncryptedCardInfo	Information of physical card to be digitized. It contains NetworkType, CardReferenceId and

Parameter	Type	Description
		byte array of Encrypted Card Info which needs to be digitized. Encrypted Card Information must be received from the Santander Server. Please refer Un-Encrypted Card Information Sample section.
decisioningData	DecisioningData	Contains data relevant for digitization decision. This is required to be passed to MDES & VTS in digitize request
emailAddress	String	Primary E-mail account address configured in mobile device. **NOTE:** • Before calling this API, every-time MPA should programmatically fetch the primary email account address configured in the mobile device. To check how to obtain the configured email account address from a mobile device, refer *Wallet Management Overview* → *Create Wallet* → *NOTE* section in the Integration Guide. • This is an Optional Parameter and CW-SDK doesn't validate the emailAddress. • This emailAddress is used by the Wallet Server to generate the digitization recommendation. Then Wallet Server send this recommendation to the ISSUER via MDES. ISSUER provides the digitization decision based on this recommendation. • This emailAddress should be same as used during the Create Wallet API call. In case, if the different emailAddress is used or primary e-mail account is changed after create wallet, then ISSUER may (or may not) give the REQUIRE_ADDITIONAL_AUTHENTICATION or DECLINED decision for the digitized card (depending upon the business). Check decision in PaymentToken for more details.

4.5.2 Response Parameters

Parameter	Type	Description
paymentToken	PaymentToken	PaymentToken (Digitized Card) information would be returned.

Parameter	Type	Description
		Refer PaymentToken section in Appendix for details of PaymentToken object.

4.5.3 Error Codes

Error Code	Description
AUTH_TOKEN_AUTHORIZATION_FAILED	ID&V token authorization failed.
WALLET_NOT_REGISTERED_WITH_TSP	Wallet is not registered with MDES/VTS.
TNC_REQUIRED	Terms and Conditions is required to digitize the card.
TNC_ACCEPTANCE_FAILED	Terms and Conditions acceptance failed on WS or MDES/VTS.
MISSING_EXPIRY_DATE	The expiry date is required for this card product but was missing.
NO_DATA_CONNECTIVITY	No data connectivity available.
PAN_INELIGIBLE_FOR_DEVICE	The PAN is not allowed to be provisioned to the device because of Issuer rules.
CARD_ALREADY_DIGITIZED	Card was already digitized previously.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
PAN_PROVISIONING_COUNT_EXCEEDED	The PAN has already been provisioned to the maximum number of devices.
DEVICE_INELIGIBLE	The device is not supported for use with MDES/VTS.
INVALID_PAN	The PAN format is not valid, or other data associated with the PAN was incorrect or entered incorrectly.
PAN_INELIGIBLE	The PAN is not in an approved account range for MDES/VTS.
INVALID_INPUT	Required input arguments are invalid.
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code.**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
PROVISION_FAILED	The provisioning of PAN to the device has failed. Please delete this card.

Error Code	Description
	Note: **When this exception received, Its responsibility of MPA to delete that payment token.**
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note:** **This exception will only occur in the case of Visa.**
AUTH_TOKEN_CREDENTIAL_MISMATCH	ID&V credentials passed during digitize call is not same as ID&V credentials used during wallet creation.
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.
DECRYPTION_FAILED	Error while decrypting the card info data received from Santander System

Notes:

- **Prior to calling this API, WL-MPA must have previously created the wallet using createWallet method and registration with associated TSP should be done using registerWith*MDES* or registerWith*VTS* method.**
- **If this API throws "PROVISION_FAILED" exception, then MPA should delete that PaymentToken by calling deletePaymentToken API.**
- **CW-SDK performs SafetyNet check asynchronously in this API call when the environment is other than DEV and WalletStatus is ACTIVE.**
- **It's responsibility of WL-MPA to provide the encrypted card information received from the Santander Server.**

4.6 digitizeByList

This method is used to perform digitization of multiple cards. WL-MPA can input list of cards to be digitized for which this API returns the processed PaymentTokens having digitization status along with display attributes for each of the given card.

Refer digitize API to get more information on card digitization process.

Alike digitize API, to check the status of digitization process of a given card, WL-MPA should look for decision value in the respective PaymentToken. Decision value as APROVED indicates that card is approved for digitation.

Decision value as REQUIRE_ADDITIONAL_AUTHENTICATION indicates Yellow path for which WL-MPA should follow the same process mentioned in digitize API call.

In case of digitization failure, either this API throws WalletSdkException or returns the list of success or failed PaymentToken. In case when this API throws WalletSdkException, error code can be retrieved by calling *WalletSdkException.getErrorCode* method. In case when this API returns the list of success or failed PaymentTokens, *PaymentToken.getDigitizationFailedReason method* will be null for successful PaymentToken and *PaymentToken.getDigitizationFailedReason* method will not be null for failed PaymentTokens. In case of failed PaymentTokens, error code can be retrieved by calling *PaymentToken.getDigitizationFailedReason.getErrorCode method*.

Check Error Codes section which is combined list of error codes returned by calling **WalletSdkException.getErrorCode** method or **PaymentToken.getDigitizationFailedReason.getErrorCode** method**.**

In that case if **PaymentToken.getDigitizationFailedReason.getErrorCode** method returns PROVISION_FAILED error code then MPA should delete respective PaymentToken by calling deletePaymentToken API. Decision value as DECLINED indicates the denial. For decision value as FAILED, WL-MPA can query the reason behind the failure by checking DigitizationFailedReason value inside the respective PaymentToken object.

4.6.1 Input Parameters

Parameter	Type	Description
authToken	AuthToken	Authorization token received from Santander ID&V after successful authentication. This object will be composed of identity token and data token.
cardInfoList	List<CardInfo>	Information of cards to be digitized. Refer CardInfo section in Appendix for details of CardInfo object.
decisioningData	DecisioningData	Contains data relevant for digitization decision. This is required to be passed to MDES & VTS in digitize request.
emailAddress	String	Primary E-mail account address configured in mobile device. **NOTE:** • Before calling this API, every-time MPA should programmatically fetch the primary email account address configured in the mobile device. To check how to obtain the configured email account address from a mobile device, refer *Wallet Management*

Parameter	Type	Description
		Overview → *Create Wallet* → *NOTE* section in the Integration Guide. • This is an optional parameter and CW-SDK doesn't validate the emailAddress. • This emailAddress is used by the Wallet Server to generate the digitization recommendation. Then Wallet Server send this recommendation to the ISSUER via MDES. ISSUER provides the digitization decision based on this recommendation. • This emailAddress should be same as used during the Create Wallet API call. In case, if the different emailAddress is used or primary e-mail account is changed after create wallet, then ISSUER may (or may not) give the REQUIRE_ADDITIONAL_AUTHENTICATION or DECLINED decision for the digitized card (depending upon the business). Check decision in PaymentToken for more details.

4.6.2 Response Parameters

Parameter	Type	Description
List of paymentTokens	List<PaymentToken>	Represents information of cards which are digitized. Refer PaymentToken section in Appendix for details of PaymentToken object.

4.6.3 Error Codes

Error Code	Description
AUTH_TOKEN_AUTHORIZATION_FAILED	ID&V token authorization failed.
WALLET_NOT_REGISTERED_WITH_TSP	Wallet is not registered with MDES/VTS.
TNC_REQUIRED	Terms and Conditions is required to digitize the card.
TNC_ACCEPTANCE_FAILED	Terms and Conditions acceptance failed on WS or MDES/VTS.
MISSING_EXPIRY_DATE	The expiry date is required for this card product but was missing.
NO_DATA_CONNECTIVITY	No data connectivity available.
PAN_INELIGIBLE_FOR_DEVICE	The PAN is not allowed to be provisioned to the device because of Issuer rules.

Error Code	Description
CARD_ALREADY_DIGITIZED	Card was already digitized previously.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
PAN_PROVISIONING_COUNT_EXCEEDED	The PAN has already been provisioned to the maximum number of devices.
DEVICE_INELIGIBLE	The device is not supported for use with MDES/VTS.
INVALID_PAN	The PAN format is not valid, or other data associated with the PAN was incorrect or entered incorrectly.
PAN_INELIGIBLE	The PAN is not in an approved account range for MDES/VTS.
INVALID_INPUT	Required input arguments are invalid.
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code.**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
PROVISION_FAILED	The provisioning of PAN to the device has failed. Please delete this card. **Note:** **When this exception received, Its responsibility of MPA to delete that payment token.**
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note:** **This exception will only occur in the case of Visa.**
AUTH_TOKEN_CREDENTIAL_MISMATCH	ID&V credentials passed during digitize call is not same as ID&V credentials used during wallet creation.
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device

Error Code	Description
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Note:

- **CW-SDK performs SafetyNet check asynchronously in this API call when the environment is other than DEV and WalletStatus is ACTIVE.**
- **Refer Notes section in digitize API for more information.**
- **In case when WalletSdkException occurs, this API will not return the List<PaymentToken>.**

4.7 digitizeByList (Using Encrypted Card Information)

This method is used to perform digitization of multiple cards using the **Encrypted Card Information**. The WL-MPA would receive the list of **Encrypted Card Information** from the Santander Server**.** WL-MPA can input list of encrypted cards (in EncryptedCardInfo) to be digitized for which this API returns the processed PaymentTokens having digitization status along with display attributes for each of the given card.

Refer digitize API to get more information on card digitization process.

Alike digitize API, to check the status of digitization process of a given card, WL-MPA should look for decision value in the respective PaymentToken. Decision value as APROVED indicates that card is approved for digitation.

Decision value as REQUIRE_ADDITIONAL_AUTHENTICATION indicates Yellow path for which WL-MPA should follow the same process mentioned in digitize API call.

In case of digitization failure, either this API throws WalletSdkException or returns the list of success or failed PaymentToken. In case when this API throws WalletSdkException, error code can be retrieved by calling *WalletSdkException.getErrorCode* method. In case when this API returns the list of success or failed PaymentTokens, *PaymentToken.getDigitizationFailedReason method* will be null for successful PaymentToken and *PaymentToken.getDigitizationFailedReason* method will not be null for failed PaymentTokens. In case of failed PaymentTokens, error code can be retrieved by calling *PaymentToken.getDigitizationFailedReason.getErrorCode method*.

Check Error Codes section which is combined list of error codes returned by calling **WalletSdkException.getErrorCode** method or **PaymentToken.getDigitizationFailedReason.getErrorCode** method**.**

In that case if **PaymentToken.getDigitizationFailedReason.getErrorCode** method returns PROVISION_FAILED error code then MPA should delete respective PaymentToken by calling deletePaymentToken API. Decision value as DECLINED indicates the denial. For decision value as FAILED, WL-MPA can query the reason behind the failure by checking DigitizationFailedReason value inside the respective PaymentToken object.

4.7.1 Input Parameters

Parameter	Type	Description
authToken	AuthToken	Authorization token received from Santander ID&V after successful authentication. This object will be composed of identity token and data token.
encryptedCardInfoList	List<EncryptedCardInfo>	List<EncryptedCardInfo> of Encrypted CardInfo which needs to be digitized. Encrypted Card Information must be received from the Santander Server.
emailAddress	String	Primary E-mail account address configured in mobile device. **NOTE:** • Before calling this API, every-time MPA should programmatically fetch the primary email account address configured in the mobile device. To check how to obtain the configured email account address from a mobile device, refer *Wallet Management Overview* → *Create Wallet* → *NOTE* section in the Integration Guide. • This is an optional parameter and CW-SDK doesn't validate the emailAddress. • This emailAddress is used by the Wallet Server to generate the digitization recommendation. Then Wallet Server send this recommendation to the ISSUER via MDES. ISSUER provides the digitization decision based on this recommendation. • This emailAddress should be same as used during the Create Wallet API call. In case, if the different emailAddress is used or primary e-mail account is changed after create wallet, then ISSUER may (or may not) give the REQUIRE_ADDITIONAL_AUTHENTICATION or DECLINED decision for the digitized card (depending upon the business). Check decision in PaymentToken for more details.
decisioningData	DecisioningData	Contains data relevant for digitization decision. This is required to be passed to MDES & VTS in digitize request.

4.7.2 Response Parameters

Parameter	Type	Description
List of paymentTokens	List<PaymentToken>	Represents information of cards which are digitized. Refer PaymentToken section in Appendix for details of PaymentToken object.

4.7.3 Error Codes

Error Code	Description
AUTH_TOKEN_AUTHORIZATION_FAILED	ID&V token authorization failed.
WALLET_NOT_REGISTERED_WITH_TSP	Wallet is not registered with MDES/VTS.
TNC_REQUIRED	Terms and Conditions is required to digitize the card.
TNC_ACCEPTANCE_FAILED	Terms and Conditions acceptance failed on WS or MDES/VTS.
MISSING_EXPIRY_DATE	The expiry date is required for this card product but was missing.
NO_DATA_CONNECTIVITY	No data connectivity available.
PAN_INELIGIBLE_FOR_DEVICE	The PAN is not allowed to be provisioned to the device because of Issuer rules.
CARD_ALREADY_DIGITIZED	Card was already digitized previously.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
PAN_PROVISIONING_COUNT_EXCEEDED	The PAN has already been provisioned to the maximum number of devices.
DEVICE_INELIGIBLE	The device is not supported for use with MDES/VTS.
INVALID_PAN	The PAN format is not valid, or other data associated with the PAN was incorrect or entered incorrectly.
PAN_INELIGIBLE	The PAN is not in an approved account range for MDES/VTS.
INVALID_INPUT	Required input arguments are invalid.
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code.**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
PROVISION_FAILED	The provisioning of PAN to the device has failed. Please delete this card. **Note:** **When this exception received, Its responsibility of MPA to delete that payment token.**

Error Code	Description
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note:** **This exception will only occur in the case of Visa.**
AUTH_TOKEN_CREDENTIAL_MISMATCH	ID&V credentials passed during digitize call is not same as ID&V credentials used during wallet creation.
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.
DECRYPTION_FAILED	Error while decrypting the card info data received from Santander System

Note:

- **CW-SDK performs SafetyNet check asynchronously in this API call when the environment is other than DEV and WalletStatus is ACTIVE.**
- **Refer Notes section in digitize API for more information.**
- **In case when WalletSdkException occurs, this API will not return the List<PaymentToken>.**

4.8 getTermsAndConditions

This method is used to get the latest Terms & Conditions from Wallet Server.

This method cannot be called before trying a first digitization as until then the wallet server does not get a valid Asset ID pointing to such Terms & Conditions. After the first digitization call, this method can be called at any time.

MDES case: Once an Asset (here Terms & Conditions) has been assigned by MDES to an Asset ID, the contents of the Asset will not change. If contents do need to change (for example, updated Terms & Conditions), they will be assigned a new Asset ID.

4.8.1 Input Parameters

Parameter	Type	Description
No Parameters required		

4.8.2 Response Parameters

Parameter	Type	Description
termsAndCondition	TermsAndCondition	Information related to Terms And Conditions.

4.8.3 Error Codes

Error Code	Description
TNC_NOT_AVAILABLE	Terms and Conditions is not available on Wallet Server
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
NO_DATA_CONNECTIVITY	No data connectivity between MPA and Wallet Server
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code.**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
~~PAN_INELIGIBLE~~	~~The PAN is not in an approved account range for MDES/VTS.~~ **Note: This error code is deprecated.**
~~INVALID_PAN~~	~~The PAN format is not valid, or other data associated with the PAN was incorrect or entered incorrectly.~~ **Note: This error code is deprecated.**
~~UNKNOWN_PAYMENT_TOKEN~~	~~PaymentToken is unknown.~~ **Note: This error code is deprecated.**

Note:

This is get request to fetch the Terms and Conditions information from WS.

4.9 requestActivationCode

This method is used to request an activation code to be sent to User. Activation code can be required in case of additional authentication required during digitization.

CW-SDK would request WS which in turn would request MDES/VTS to generate and send the activation code to User.

The digitize Service supports different methods of activation, for example: using a user-entered Activation Code, using an Issuer's mobile app, or by contacting customer service. The Issuer determines which activation method(s) are available to the Cardholder, who is prompted to select one of the supported activation methods.

This method is used only for the user-entered activation code method of activation. It requires activation code to be passed by WL-MPA which then would be send to WS for further processing with MDES/VTS.

4.9.1 Input Parameters

Parameter	Type	Description
paymentTokenId	String	Unique PaymentToken reference ID
activationMethod	ActivationMethod	Activation Method to activate PaymentToken. Refer Appendix for details about ActivationMethod object.

4.9.2 Response Parameters

Parameter	Type	Description
No Parameters Required.		

4.9.3 Error Codes

Error Code	Description
~~ACTIVATION_ERROR~~	~~It could be due to other reason like session expired.~~ Note: This error code has been deprecated.
ACTIVATION_PERIOD_EXPIRED	Activation period is expired. Refer Notes below
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_PAYMENT_TOKEN_STATUS	The token is in an invalid status for the requested operation.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.

Error Code	Description
NO_DATA_CONNECTIVITY	No data connectivity between MPA and Wallet Server
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code.**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
INVALID_ACTIVATION_METHOD	The activation method could not be found.
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note:** **This exception will only occur in the case of Visa.**
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Note:

Once an Activation Code has been generated, it will be valid for a limited time period, after which the code will expire. Once a code expires, WL-MPA can request for a new Activation Code, as long as the activation period has not expired. Calling this method again will not cause the Activation Code to be regenerated nor extend the validity period of the Activation Code.

4.10 activate

This method is used to activate the PaymentToken (Digitized Card) using an activation code.

The digitize Service supports different methods of activation, for example: using a user-entered Activation Code, using an Issuer's mobile app, or by contacting customer service. The Issuer determines which activation method(s) are available to the Cardholder, who is prompted to select one of the supported activation methods.

This method is used only for the user-entered activation code method of activation. It requires activation code to be passed by WL-MPA which then would be send to WS for further processing with MDES/VTS.

4.10.1 Input Parameters

Parameter	Type	Description
paymentTokenId	String	Unique PaymentToken reference ID
activationCode	String	Activation code required for activating the PaymentToken

4.10.2 Response Parameters

Parameter	Type	Description
activationResult	ActivationResult	Activation Result Refer Appendix for various ActivationResult values

4.10.3 Error Codes

Error Code	Description
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_PAYMENT_TOKEN_STATUS	The token is in an invalid status for the requested operation.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.
NO_DATA_CONNECTIVITY	No data connectivity between WL-MPA and Wallet Server
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code.**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note:** **This exception will only occur in the case of Visa.**
DEVICE_NOT_SUPPORTED	Device not supported

Error Code	Description
OS_NOT_SUPPORTED	OS is not supported
~~ACTIVATION_ERROR~~	~~It could be due to other reason like session expired.~~ **Note: This error code is deprecated.**
~~INVALID_WORKFLOW~~	~~There is an error in workflow.~~ ~~For example, card activation is called before requesting for activation code, SUSPEND or RESUME called on INACTIVE Token and other errors.~~ **Note: This error code is deprecated.**
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device.
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Note:

Note that the user is only given a limited number of attempts to enter a correct Activation Code (typically 3 attempts), after which the Activation Code becomes invalid. In this event, it may be possible to request a new Activation Code directly from the Issuer via customer service or otherwise. This is dependent on individual Issuer implementation and out of scope of this API.

Reference use cases:

1. Add Card
   - Activate – Yellow path digitization

4.11 terminateWallet

This method is used to terminate wallet instance on this device. A valid PIN or secured device unlock is required to make terminate wallet instance process.

CW-SDK would make a Wallet Server request for termination process. On Wallet Server, based on the provided PIN, authentication would be performed. If authenticated, WS would connect with other systems to effect the termination of Wallet instance on those systems.

Upon a successful termination response from WS, the CW-SDK deletes all existing Transaction Credentials and mark the wallet instance on CW-SDK as terminated.

Note:

WL-MPA can leverage retrieveTerminateReason API from WalletLifeCyclePreferenceHelper utility class to know the wallet terminate reason in detail.

4.11.1 Input Parameters

Parameter	Type	Description
walletPIN	Byte [ ]	Wallet PIN. Conditional – Required if PIN is managed by WL-MPA

4.11.2 Response Parameters

Parameter	Type	Description
result	Boolean	TRUE on successful Wallet termination.

4.11.3 Error Codes

Error Code	Description
~~INVALID_WALLET_PIN~~	~~Invalid Wallet PIN~~ **Note: This error code is deprecated.**
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
NO_DATA_CONNECTIVITY	No data connectivity between MPA and Wallet Server
COMMUNICATION_ERROR	Error while communicating with remote server. **Note: May get various system error response from server. All would be collated under this error code**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
DEVICE_NOT_SUPPORTED	Device not supported
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Reference use cases:

1. Terminate Wallet

4.12 getWalletInfo

This method is used to fetch the information of Wallet instance stored locally in CW-SDK.

CW-SDK manages the life-cycle of Wallet instance associated with it. This method returns information of Wallet instance at a requested time.

WalletInfo would contain status of Wallet along few other information.

In the case of WalletStatus value returned as TERMINATED, WL-MPA can leverage retrieveTerminateReason API from WalletLifeCyclePreferenceHelper utility class to know the wallet TerminateReason in detail.

Note:

The getWalletInfo API will not provide value of NewAvailableVersionNumber inside WalletInfo response even if there is a new version available. It will be provided only when walletUpdated notification is given to WL-MPA.

4.12.1 Input Parameters

Parameter	Type	Description
Not Input Parameters		

4.12.2 Response Parameters

Parameter	Type	Description
walletInfo	WalletInfo	Information of Wallet instance stored locally with Wallet. Refer WalletInfo object in Appendix to get details of the object

Reference use cases:

1. On Application Launch

4.13 getTokenForOperation

This method is used to calculate the hash of operation and parameters passed by WL-MPA.

4.13.1 Input Parameters

Parameter	Type	Description
operation	Operation	Type of Operation to be WL-MPA wants to execute.
operationParams	OperationParams	Parameters used in token generation. List of parameters would vary based on the operation. Refer OperationParams section in Appendix

4.13.2 Response Parameters

Parameter	Type	Description
hashedToken	String	Hashed Token of the passed values.

4.13.3 Error Codes

Error Code	Description
INVALID_INPUT	Passed parameters does not match or are invalid for the operation.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
DEVICE_NOT_SUPPORTED	Device not supported

4.14 setWalletNotificationListener

This method is used to set the Wallet notification handler.

During various wallet and PaymentToken life-cycle changes, CW-SDK would send various notification events to the registered handler passed in this method.

WL-MPA would have to register handler to receive wallet and card notifications from CW-SDK.

List of all the notifications events sent by CW-SDK are listed in separate chapter in the document.

4.14.1 Input Parameters

Parameter	Type	Description
walletNotificationListener	WalletNotificationListener	Wallet notification handler

4.14.2 Response Parameters

Parameter	Type	Description
Not Response Parameters		

Reference use cases:

1. Launch Wallet

4.15 setTransactionNotificationListener

This method is used to set the Transaction notification handler.

During transaction, CW-SDK would send various notification events to the registered handler passed in this method.

WL-MPA would have to register handler to receive transaction notifications from CW-SDK.

List of all the notifications events sent by CW-SDK are listed in separate chapter in the document.

4.15.1 Input Parameters

Parameter	Type	Description
transactionNotificationListener	TransactionNotificationListener	Transaction notification handler

4.15.2 Response Parameters

Parameter	Type	Description
Not Response Parameters		

Reference use cases:

1. Initiate Transaction (Tap n Pay)

4.16 setProvisionFailedNotificationListener

This method is used to set the Provision Failed Notification listener.

During the provisioning process for any Payment Card(s), if URPay SDK is not able to complete the provisioning process, than this notification will be raised by CW-SDK.

WL-MPA would have to register a listener to receive this notifications from CW-SDK via this API.

4.16.1 Input Parameters

Parameter	Type	Description
provisionFailedNotificationListener	ProvisionFailedNotification Listener	Listener to get event of Provisioning process failure for Payment Card(s)

4.16.2 Response Parameters

Parameter	Type	Description
Not Response Parameters		

Reference use cases:

1. Digitize Card

4.17 acceptSafetyNetFailedRisk

This method is used to provide user acceptance in the case when the SafetyNet check detects the device as potentially tempered. In this case, CW-SDK will provide safetyNetFailed notification along with error code as SAFETYNET_USER_CONSENT_REQUIRED. It is the responsibility of WL-MPA to show a proper message to the user and accept the consent.

Once user will accept the risk, WL-MPA needs to call this API so that wallet server will save the user consent and user will not get same error callback later on.

4.17.1 Input Parameters

Parameter	Type	Description
No input parameters		

4.17.2 Response Parameters

Parameter	Type	Description
No Response Parameters		

4.17.3 Error Codes

Error Code	Description
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code**
DEVICE_NOT_SUPPORTED	Device not supported.
NO_DATA_CONNECTIVITY	No data connectivity available.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device

Reference use cases:

- Launch Wallet
- Wallet Notifications - safetyNetFailed (WalletSdkException walletSDKException).

4.18 updateRnsRegistrationIds

This method is used to update the RNS registration ID of Wallet Server, CES and MDES.

The WL-MPA would call this API only in below scenario:

- In the case when FCM determines that the tokens need to be changed, the WL-MPA would get the onNewToken callback. After getting this callback, the WL-MPA should send the updated token to the CW-SDK using this API.

4.18.1 Input Parameters

Parameter	Type	Description

authToken	AuthToken	Authorization token received from Santander ID&V after successful authentication. This object will be composed of identity token and data token
walletServerRnsRegistrationId	String	RNS Identifier registered by the WL-MPA for Wallet Server.
cesRnsRegistrationId	String	RNS Identifier registered by the WL-MPA for CES. Conditional – Required if MPA country has CES system
mdesRnsRegistrationId	String	RNS Identifier registered by the WL-MPA for MDES. Conditional – Required if the WL-MPA previously registered with MDES else Optional.

4.18.2 Response Parameters

Parameter	Type	Description
No Response		

4.18.3 Error Codes

Error Code	Description
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code**
DEVICE_NOT_SUPPORTED	Device not supported.
NO_DATA_CONNECTIVITY	No data connectivity available.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
AUTH_TOKEN_AUTHORIZATION_FAILED	ID&V token authorization failed.
AUTH_TOKEN_CREDENTIAL_MISMATCH	ID&V credentials passed during registerWithMdes call is not same as ID&V credentials used during wallet creation.
OS_NOT_SUPPORTED	OS is not supported.
INVALID_INPUT	Required input arguments are invalid.
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet

Error Code	Description
	PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Note:

- **To check how to obtain the registration id for multiple firebase projects, Please refer Firebase Cloud Messaging Configuration section in the integration guide for more details.**
- **In a case if this API throws any exception then MPA should retry until MPA receives success response.**

5 PaymentToken Services

Reference use cases:

1. Launch Wallet
2. Register Wallet
3. Digitize Card

In the CW SDK API, PaymentToken services apply only to DPAN being provisioned in the device. Those services don’t apply to static token or server DSRP token.

5.1 getPaymentTokens

This method is used to get all the PaymentTokens (Digitized Card) stored locally within CW-SDK. It returns a list of PaymentToken objects.

PaymentTokens would be persisted and managed locally by CW-SDK and can be fetched anytime by WL-MPA to display it on screens or for any other purpose. The stored PaymentToken information does not contain any critical information hence authentication is not required to access this information.

5.1.1 Input Parameters

Not required

5.1.2 Response Parameters

Parameter	Type	Description
paymentTokens	List<PaymentToken>	List of token information would be returned. Returns empty list if no tokens are available. Refer PaymentToken object in Appendix for details.

5.1.3 Error Codes

Error Code	Description
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Note:

- **In case if PaymentToken.getDigitizationFailedReason() is not null that means, Card digitization is failed. In that case, if DigitizationFailedReason.getErrorCode() returns PROVISION_FAILED error code then MPA should delete respective PaymentToken by calling deletePaymentToken API.**

Reference use cases:

1. Display dashboard
2. Card Setup

5.2 deletePaymentToken

This method is used to delete a PaymentToken (Digitized Card).

CW-SDK would send a delete PaymentToken request to WS which would then request MDES for the deletion of requested PaymentToken.

MDES will coordinate the deletion of the Token and notify any relevant parties that the Tokens have now been removed.

On successful response from WS, CW-SDK would remove the PaymentToken from the local storage. Further, it would also delete all the transaction keys for the deleted PaymentToken.

5.2.1 Input Parameters

Parameter	Type	Description
paymentTokenId	String	Unique PaymentToken reference ID

5.2.2 Response Parameters

Parameter	Type	Description
No Parameters required		

5.2.3 Error Codes

Error Code	Description
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_PAYMENT_TOKEN_STATUS	The token is in an invalid status for the requested operation.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.
NO_DATA_CONNECTIVITY	No data connectivity between MPA and Wallet Server
COMMUNICATION_ERROR	Error while communicating with remote server. **Note: May get various system error response from server. All would be collated under this error code.**

Error Code	Description
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note: This exception will only occur in the case of Visa.**
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Note:

If the default PaymentToken is deleted, CW-SDK would not set any other PaymentToken as default. WL-MPA is expected to set the default PaymentToken.

Reference use cases:

1. Delete Card

5.3 suspendPaymentToken

This method is used to temporarily suspend a PaymentToken (Digitized Card).

CW-SDK would send a suspend PaymentToken request to WS which would then request MDES for the suspension of requested token. WS can also inform Santander systems for the PaymentToken (card) suspension.

On response, CW-SDK marks the PaymentToken as suspended.

Suspended PaymentToken can be unsuspended using resumePaymentToken method.

5.3.1 Input Parameters

Parameter	Type	Description
paymentTokenId	String	Unique PaymentToken reference ID to be suspended

5.3.2 Response Parameters

Parameter	Type	Description
No Parameters required		

5.3.3 Error Codes

Error Code	Description
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_PAYMENT_TOKEN_STATUS	The token is in an invalid status for the requested operation.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.
NO_DATA_CONNECTIVITY	No data connectivity between MPA and Wallet Server
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:** **May get various system error response from server. All would be collated under this error code**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note: This exception will only occur in the case of Visa.**
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Reference use cases:

1. Suspend Card

5.4 resumePaymentToken

This method is used to resume previously suspended PaymentToken (Digitized Card).

CW-SDK would send a resume PaymentToken request to WS which would then request MDES for resuming the PaymentToken. WS can also inform Santander systems for the PaymentToken (card) resume.

On successful response, CW-SDK updates the payment token status as active.

5.4.1 Input Parameters

Parameter	Type	Description
paymentTokenId	String	Unique PaymentToken reference ID
walletPIN	Byte [ ]	Wallet PIN for authenticating the User. Conditional – Required if PIN is managed by WL-MPA

5.4.2 Response Parameters

Parameter	Type	Description
No Parameters required		

5.4.3 Error Codes

Error Code	Description
~~INVALID_WALLET_PIN~~	~~Invalid Wallet PIN~~ **Note: This error code is deprecated.**
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_PAYMENT_TOKEN_STATUS	The token is in an invalid status for the requested operation.
RESUME_PAYMENT_TOKEN_FAILED	Resume of Payment Token failed on WS due to some specific reason. For example, Payment Token suspended by Issuer cannot be resumed using this API.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.
NO_DATA_CONNECTIVITY	No data connectivity between MPA and Wallet Server
COMMUNICATION_ERROR	Error while communicating with remote server. **Note:**

Error Code	Description
	May get various system error response from server. All would be collated under this error code
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note: This exception will only occur in the case of Visa.**
DEVICE_NOT_SUPPORTED	Device not supported
OS_NOT_SUPPORTED	OS is not supported
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Reference use cases:

1. Un-suspend Card

5.5 setDefaultPaymentToken

This method is used to set the default PaymentToken (Digitized Card) to use for Tap n Pay.

CW-SDK would mark the requested PaymentToken as default locally which will be used as default for any Tap n Pay transaction.

5.5.1 Input Parameters

Parameter	Type	Description
paymentTokenId	String	Unique PaymentToken reference Id to be marked as default

5.5.2 Response Parameters

Parameter	Type	Description
No Parameters required		

5.5.3 Error Codes

Error Code	Description
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_PAYMENT_TOKEN_STATUS	The token is in an invalid status for the requested operation.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Notes:

This will be used as the default NFC card i.e. the default card for contactless payments.

Reference use cases:

1. Set Default Card

5.6 getDefaultPaymentToken

This method is used to get the default PaymentToken which was previously set as default by WL-MPA. It returns the PaymentToken information of the default card set.

CW-SDK would maintain the status of the default PaymentToken locally.

5.6.1 Input Parameters

No parameters required

5.6.2 Response Parameters

Parameter	Type	Description
paymentToken	PaymentToken	Returns the default PaymentToken set locally. This would be NULL if there is no default payment token set.

5.6.3 Error Codes

Error Code	Description
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN,

Error Code	Description
	PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

5.7 getRemainingTransactionKeys

This method is used to fetch the number of unused transaction keys for a given PaymentToken.

CW-SDK has the ability to control the number of valid transaction keys and manage the minimum stock required performing replenishment when possible.

For Mastercard PaymentToken this API would return number of SUKs for that token. For Visa PaymentToken this API would return LUK limits for that token.

5.7.1 Input Parameters

Parameter	Type	Description
paymentTokenId	String	PaymentToken identifier for which remaining transaction key is required.

5.7.2 Response Parameters

Parameter	Type	Description
transactionKeys	Integer	Conditional – If Mastercard Token, it will return number of SUKs (Single Use Key). If Visa Token, it will return LUK (Limited Use Key) limits. Only NOT (Number of Transactions) would be returned as part of LUK limits. Even if this API returns 0 (for VISA tokens only), Transaction should be allowed. Transaction may still succeed even when the thresholds are exceeded. It is up to the issuer to decide if the payment should be declined/approved.

5.7.3 Error Codes

Error Code	Description
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_PAYMENT_TOKEN_STATUS	The token is in an invalid status for the requested operation.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.

Error Code	Description
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Note:

This API gives the remaining transaction keys for Mastercard specific cards. But, in the case of Visa specific cards, the remaining transaction keys cannot be determined because VISA uses LUK (Limited Use Keys). Therefore, in the case of Visa cards, MPA should take care of above behavior. However, MPA can still perform the transaction, where LUK replenishment will be done internally by CWSDK.

6 Transaction Services

These are transaction services which are published by CW-SDK to do transaction related activities.

6.1 initiateTransaction

This method is used to initiate transaction with given PaymentToken (Card).

CW-SDK would perform the initial velocity checks and would raise a *onPerformCVM* event/callback if CVM is required as per the velocity checks.

CW-SDK would fetch the Wallet PIN from local storage (whitebox) and use that for further transaction processing.

6.1.1 Input Parameters

Parameter	Type	Description
paymentTokenId	String	PaymentToken with which the transaction is to be done

6.1.2 Response Parameters

Parameter	Type	Description
No Response		

6.1.3 Error Codes

Error Code	Description
DEVICE_NOT_SUPPORTED	Device not supported
NFC_DISABLED	NFC is disabled.
APP_NOT_DEFAULT_FOR_PAYMENT	Application is not selected as default for payment in tap & pay device settings
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_PAYMENT_TOKEN_STATUS	The token is in an invalid status for the requested operation.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.
PAYMENT_TOKEN_EXHAUSTED	No more transaction credentials available for making payment.
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

Reference use cases:

1. Transaction Services

Note:

After user taps on the terminal, CW-SDK would raise all the relevant transaction notifications mentioned at section 7.2.

6.2 cancelTransaction

This method is used to cancel the transaction which was initiated using *initiateTransaction* API.

6.2.1 Input Parameters

Parameter	Type	Description
No input parameters		

6.2.2 Response Parameters

Parameter	Type	Description
No Response		

Note:

Transaction will be cancelled automatically in below cases:

- **When some success or failure arises after tapping device on POS.**
- **Payment token gets deleted.**
- **CW-SDK gets re-initialized again.**

6.3 getTransactionHistory

This method is used to fetch the list of transactions using the given PaymentToken that are performed from this device. It returns all the transaction history received from TDS for requested PaymentToken.

CW-SDK does not store transaction history. It makes a request to TDS to get the transaction history details for a PaymentToken.

In the case of PaymentToken having a state as Suspended, this API returns response as below:

For Mastercard - This API provides transaction history.

For VISA – This API throws exception having PAYMENT_TOKEN_SUSPENDED_FOR_TRANSACTION_HISTORY error code.

6.3.1 Input Parameters

Parameter	Type	Description
paymentTokenId	String	PaymentToken ID for which transaction history is required.
walletPIN	Byte []	PIN to authenticate User. Conditional – Required if PIN is managed by WL-MPA

6.3.2 Response Parameters

Parameter	Type	Description
transactionHistory	List<TransactionDetails>	List of transaction information if available, else it will return empty list.

6.3.3 Error Codes

Error Code	Description
~~INVALID_WALLET_PIN~~	~~Invalid Wallet PIN~~ **Note: This error code is deprecated.**
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.
ERROR_FETCHING_TRANSACTION_HISTORY	Error while fetching transaction history information from Server.
NO_DATA_CONNECTIVITY	No data connectivity between MPA and Wallet Server
COMMUNICATION_ERROR	Error while communicating with remote server. **Note: May get various system error response from server. All would be collated under this error code**
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
PAYMENT_TOKEN_SUSPENDED_FOR_TRANSACTION_HISTORY	The token is in suspended status for fetching transaction history. **Note: This exception will only occur in the case of Visa.**
PAYMENT_TOKEN_INACTIVE_FOR_TRANSACTION_HISTORY	The token is in inactive status for fetching transaction history.

Error Code	Description
	Note: This exception will only occur in the case of Visa.
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. **Note: This exception will only occur in the case of Visa.**
DEVICE_NOT_SUPPORTED	Device not supported
APP_INSTANCE_TERMINATED	Application instance is terminated.
DEVICE_IS_ROOTED	Wallet server found rooted device
RESET_REQUIRED	This error would occur in cases when URPay SDK has wiped out data i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

7 Notifications

This section lists all the notifications/events that are sent by CW-SDK to WL-MPA during various Wallet and Card life-cycle changes and also during the in-store payment by Tap & Pay.

7.1 Wallet Notifications - WalletNotificationListener

These notifications(events) are used by CW-SDK to notify the WL-MPA of significant Wallet and Card (PaymentToken) state change updates, such as when the Card (PaymentToken) is activated, suspended, unsuspended or deleted; or when information about the PaymentTokens such as replenishment completed and card is ready to pay.

It would be triggered once the underlying payment SDK finishes all the operations related to the given operations (for example, once PaymentToken is provisioned and transaction keys are replenished it would raise a notification).

WL-MPA would have to register handler to receive wallet and card notifications from CW-SDK. Below are the list of notifications/events under wallet notifications.

7.1.1 walletUpdated

This event would be called when the CW-SDK identifies the change in wallet state.

CW-SDK implicitly performs sync operation with Wallet Server at various events to synchronize the Wallet/PaymentTokens state between CW-SDK and Wallet Server. Post this operation, if CW-SDK identifies any change in state of Wallet then it will raise this notification.

7.1.1.1 Input Parameters

Parameter	Type	Description
walletInfo	WalletInfo	Few of the wallet information. Refer WalletInfo section in Appendix for information.

7.1.2 paymentTokenUpdated

This event would be called when the CW-SDK identifies the change in PaymentToken information.

CW-SDK implicitly performs sync operation with Wallet Server at various events to synchronize the Wallet/PaymentTokens state between CW-SDK and Wallet Server. Post this operation, if CW-SDK identifies any change in state of PaymentTokens (for example, PaymentToken suspended, activated, deleted, and other states of payment tokens) then it will raise this notification.

7.1.2.1 Input Parameters

Parameter	Type	Description
paymentTokens	List<PaymentToken>	Information of all the updated PaymentTokens.

Parameter	Type	Description
		Refer Appendix for details of PaymentToken structure.

Notes

This event would not be raised when WL-MPA makes *synchronize* method call to synchronize the data. In this case, updated information would be received in *synchronize* method response. Refer synchronize method API description for further details.

This event would not be raised in scenarios when Payment Token update operation such as suspend, resume, delete, etc. is initiated from WL-MPA (by User) using respective methods/APIs.

In scenario when Card is updated (suspended, resumed, deleted) in backend, then a push notification would be received by CW-SDK for such updates. On this push notification, CW-SDK would make the necessary business and would raise this event.

7.1.3 paymentToken Replenished

This event would be called when CW-SDK completes initial replenish after PaymentToken is activated and PaymentToken is ready for payment.

CW-SDK will internally perform replenishment operation after PaymentToken is personalized and gets activated. Post successful replenishment CW-SDK will raise this event to inform WL-MPA about the PaymentToken is ready for the transaction.

7.1.3.1 Input Parameters

Parameter	Type	Description
paymentToken	PaymentToken	Information of payment Token which is ready for payment after replenish operation

Notes

This event will be call only after first time (Initial) replenishment.

This event will be called only for Mastercard Cards.

7.1.4 safetyNetFailed

This event would be raised when CW-SDK found the device as potentially tempered and user consent/acceptance is required while SafetyNet checks during CW SDK initialization.

- When the device is found potentially tempered and user consent/acceptance is required about the risk, CW-SDK will notify MPA via safetyNetFailed notification having walletSDKException as a parameter. In the case when this parameter will have error code value as SAFETYNET_USER_CONSENT_REQUIRED, WL-MPA needs to take user acceptance by showing proper Information to the user. When the user provides his consent/acceptance, MPA must call acceptSafetyNetFailedRisk CW-SDK API.

- In the case when the WalletSDKException is having error code value as SAFETYNET_DEVICE_POSSIBLY_TEMPERED, user consent/acceptance is not required. In this case, MPA needs to show proper information to the user.

7.1.4.1 Input Parameters

Parameter	Type	Description
walletSDKException	WalletSdkException	CW-SDK will send walletSDKException with below error codes. • SAFETYNET_USER_CONSENT_REQUIRED • SAFETYNET_DEVICE_POSSIBLY_TEMPERED

Note:

Wallet gets terminated after maximum number (as defined during tenant onboarding) of SAFETYNET_DEVICE_POSSIBLY_TEMPERED error occurrences. In this case, CW-SDK will provide walletUpdated notification with wallet status as TERMINATED.

7.1.5 paymentTokenDigitized

This event would be raised each time whenever any Payment Card digitization is successful while calling the digitizeByList or digitizeByList (Using Encrypted Card Information) API. This callback can be leveraged if WL-MPA needs to update the UI when any card get successfully digitized while calling the digitizeByList or digitizeByList (Using Encrypted Card Information)API call.

7.1.5.1 Input Parameters

Parameter	Type	Description
paymentTokens	PaymentToken	Information of all the digitized PaymentTokens. Refer Appendix for PaymentToken structure details.

Note:

This event would not be raised in-case of WL-MPA makes digitize or digitize (Using Encrypted Card Information) API call to digitize a single Payment Card.

7.2 Transaction Notifications - TransactionNotificationListener

Notifications sent from CW-SDK to WL-MPA during transaction.

7.2.1 getConsent

CW-SDK would send this callback event to get the consent of WL-MPA (user) which is required to decide whether transaction must be allowed to proceed or should be declined.

Based on the business requirement, WL-MPA can perform various checks such as device secure/unsecure lock is enabled or device screen is lit etc. and return the consent (true/false) to perform the transaction.

7.2.1.1 Output Parameters

Parameter	Type	Description
consent	Boolean	Flag indicates whether transaction can proceed or declined.

Note:

If return false, transaction will be terminated and *onTransactionError* will be raised with *UNKNOWN_ERROR*.

```
getAuthenticationState
```

CW-SDK would send this callback event to get the authentication status of WL-MPA which is required to perform the transaction.

WL-MPA would maintain the authentication state and would return the relevant information when this event is called.

7.2.1.2 Output Parameters

Parameter	Type	Description
authenticationState	Authentication State	Authentication state of User. Refer Appendix for Authentication State object.

Note:

This is a synchronous call. CW-SDK would wait to do further processing until the method returns with the response.

Based on the authentication information passed in response of this event and after applying certain velocity rules, CW-SDK would identify if User authentication is required to perform the transaction. In case, when user authentication is required to proceed with transaction, onPerformCVM event/notification would be raised by CW-SDK.

For case when CVM is always required, *AuthenticationState#isAuthenticated* can be returned as *false* by MPA. If done so, CW-SDK would identify the need of user authentication while verifying for CDCVM & velocity checks and would raise onPerformCVM notification.

7.2.2 onPerformCVM

CW-SDK would send this event when user authentication is required to perform the transaction.

Before initiating the transaction, CW-SDK checks validity of user authentication and also performs various other velocity check rules. While performing this checks if it identifies the requirement of user authentication then it raises this event.

Caution: It is not guaranteed that this notification API will always provide correct terminal type information because of limitations on EMVCo specifications. The notification API has been optimized for a CVM List 00 00 00 00 00 00 00 00 02 03 1F 03 or 00 00 00 00 00 00 00 00 42 03 1F 03 (tag 8E, "online PIN", "No CVM Supported") and to return terminal Type as V2 terminal either In the case when the terminal is a V2 or the terminal is a V3 but without CDCVM support (Where behavior is similar to V2 terminal). It is recommended to WL-MPA to show proper message to user in any case.

Note: OnPerformCVM notification without any input parameter is now deprecated and will not be called by CW-SDK. It is highly recommended to use onPerformCVM with TransactionContext as input parameter API instead.

7.2.2.1 Input Parameters

Parameter	Type	Description
transactionContext	TransactionContext	TransactionContext object.

Note:

After performing CVM, device is required to tap again to POS to proceed with transaction. When the user taps again on the terminal, all the events (viz. *onTransactionStarted*, *getAuthenticationState*, etc.) would be raised again by CW-SDK.

7.2.3 onTransactionStarted

This event is called when the transaction is about to start.

CW-SDK would send a notification when contactless is ready and communication with reader has started. On this notification, WL-MPA can show the relevant screens to display it to user that transaction has started.

Note:

This is an informative event to let WL-MPA know that the contactless is ready and payment would be about to start.

7.2.4 onTransactionCompleted

This event is called when the transaction is completed.

7.2.4.1 Input Parameters

Parameter	Type	Description
transactionInfo	Transaction Info	Transaction Info object. Refer Appendix 8.1.6

Notes:

Note that this event informs that the transaction (communication with contactless reader) is completed. Once the transaction is processed, a separate event would be sent to WL-MPA which is onTransactionProcessed event.

7.2.5 onTransactionError

This event is called when there is error in transaction.

7.2.5.1 Input Parameters

Parameter	Type	Description
transactionError	TransactionError	Error message composed of error code

7.2.5.2 Error Codes

Error Code	Description
NO_CARD_ACTIVE_FOR_PAYMENT	This error code can occur in two scenarios: 1. The user tried to pay, but there was no default payment card found. 2. The user tried to pay with a default card which was suspended earlier. In second scenario MPA need to take care and show message appropriate message to the user when a default card is also in suspended state.
DEVICE_LEFT_NFC_FIELD	The device left the NFC field while payment transaction was in progress.
TRANSACTION_CONTEXT_CONFLICT	Second Tap was performed by user but terminal had changed the amount or currency since the First Tap.
UNKNOWN_ERROR	Transaction failed due to reason not known or condition of use is not satisfied.
PAYMENT_TOKEN_EXHAUSTED	No more transaction credentials available for making payment.
TRANSACTION_APPLICATION_DECLINED	Transaction is aborted due to various reasons like consent not given, wallet state is not ACTIVE and device is rooted etc.

Note:

In case when default PaymentToken is not set with CW-SDK and Tap n Pay is done by the User then this event would be raised by CW-SDK. MPA would display card selection screen and will ask the user to select the card (PaymentToken).

On user card selection, MPA would use *initiateTransaction* method to proceed with the Payment.

7.2.6 onTransactionProcessed

This event is called when the transaction information (transaction receipt) has been received from TDS and it matches the previously completed transaction. This events provides the transaction receipt as transaction object. Status of the transaction (which is an attribute of transaction object) could be either AUTHORIZED, DECLINED, CLEARED, or REVERSED.

7.2.6.1 Input Parameters

Parameter	Type	Description
transactionDetails	Transaction Details	Transaction Details object.

7.3 Provision Failed Notification - ProvisionFailedNotificationListener

This notification (event) is used by CW-SDK to notify the WL-MPA when the provisioning process for the payment card(s) is failed.

7.3.1 onProvisionFailed

It would be triggered once the underlying URPay SDK is not able to provision the given payment card(s). Because the provisioning process is managed by URPay SDK, there is no way CW-SDK can re-initiate the provisioning process for that card.

When WL-MPA receives such notification, the only way to recover from this state is to delete the payment card(s) using the deletePaymentToken API and try to digitize that card again.

In case if WL-MPA ignores this notification, the other possible way to check the provisioning status of a card is via getPaymentTokens API. If this API returns a PaymentToken object having status value as INACTIVE for some long time, it might happen that the provisioning of that card is failed or not yet started. Also for this case, the only way to recover from this state is to delete the payment card(s) using the deletePaymentToken API and try to digitize that card again.

7.3.1.1 Input Parameters

Parameter	Type	Description
paymentTokens	List<PaymentToken>	PaymentTokens for whom the provisioning process is failed. Refer Appendix for details of PaymentToken structure.

7.4 NotificationHelper

Notification helper provides utility APIs to help WL-MPA handle notification. These APIs would be used by WL-MPA whenever WL-MPA receives push notification.

7.4.1 hasCWSPushContent

This API identifies if the notification contains CWS push content or not. This would be used by WL-MPA when it receives a notification. You can retrieve CWS push content using getCWSPushContent API.

7.4.1.1 Input Parameters

Parameter	Type	Description
remoteMessage	RemoteMessage	Push notification information

7.4.1.2 Response Parameters

Parameter	Type	Description
result	Boolean	TRUE/FALSE value whether push content has CWS content or not

7.4.2 hasMDESPushContent

This API identifies if the notification contains MDES push content or not. This would be used by WL-MPA when it receives a notification. You can retrieve MDES push content using getMDESPushContent API.

7.4.2.1 Input Parameters

Parameter	Type	Description
remoteMessage	RemoteMessage	Push notification information

7.4.2.2 Response Parameters

Parameter	Type	Description
result	Boolean	TRUE/FALSE value whether push content has MDES content or not

7.4.3 getCWSPushContent

This API retrieves CWS push content from the given remote message. This would be used by WL-MPA when it identifies the received notification has CWS push content using hasCWSPushContent.

7.4.3.1 Input Parameters

Parameter	Type	Description
remoteMessage	RemoteMessage	Push notification information

7.4.3.2 Response Parameters

Parameter	Type	Description
pushContent	String	CWS push content.

7.4.4 getMDESPushContent

This API retrieves MDES push content from the given remote message. This would be used by WL-MPA when it identifies the received notification has MDES push content using hasMDESPushContent.

7.4.4.1 Input Parameters

Parameter	Type	Description
remoteMessage	RemoteMessage	Push notification information

7.4.4.2 Response Parameters

Parameter	Type	Description
pushContent	String	MDES push content.

8 Wallet Life Cycle Helper Services - WalletLifeCyclePreferenceHelper

This section depicts various utility methods. It helps WL-MPA to get wallet termination reason as well as reset the value of terminate reason.

Use-Case:

Terminate Wallet

8.1 getInstance

This method is used to retrieve the singleton instance of WalletLifeCyclePreferenceHelper object that was created and initialized during initializeSDK method.

This method would throw an exception if it is called before initializing the SDK.

8.1.1 Input Parameters

Parameter	Type	Description
No Input Parameters required		

8.1.2 Response Parameters

Parameter	Type	Description
walletLifeCyclePreferenceHelper	WalletLifeCyclePreferenceHelper	Returns WalletLifeCyclePreferenceHelper instance object.

8.1.3 Error Codes

Error Code	Description
SDK_NOT_INITIALIZED	If CWSDK is not initialized. **Note: CWSDK can be initialized using initializeSDK method.**

8.2 retrieveTerminateReason

This method is used to get the wallet termination reason in case of wallet is terminated.

8.2.1 Input Parameters

Parameter	Type	Description
No Input Parameters required		

8.2.2 Response Parameters

Parameter	Type	Description
terminateReason	TerminateReason	Returns TerminateReason instance object.

8.3 reset

This method is used to reset wallet terminate reason.

WL-MPA must always call this method once walletUpdated event is handled successfully.

8.3.1 Input Parameters

Parameter	Type	Description
No Input Parameters required		

8.3.2 Response Parameters

Parameter	Type	Description
No Response Parameters required		

8.4 enableFileBasedLogging

This method is used to enable File Based logging for CW-SDK.

WL-MPA must always call this method using CW-SDK in Debug mode. Calling this method on CW-SDK release mode will result in exception.

WL-MPA can locate the logs files at below given path:

- **Android > data > "package name of WL-MPA" >.. > CWSDK_Logs >cwsdk_logs.txt.**

8.4.1 Input Parameters

Parameter	Type	Description
context	Context	Android application context.
loggingProperties	LoggingProperties	Instance of LoggingProperties which defines configurations for file based loggings.

8.4.2 Response Parameters

Parameter	Type	Description
No Response Parameters required		

8.4.3 Error Codes

Error Code	Description
FILE_LOGGING_NOT_ALLOWED_IN_RELEASE_MODE	Enabling file based logging is not allowed in release mode of CW-SDK. **Note: WL-MPA must use CW-SDK in Debug mode to use file based Logging.**

Error Code	Description
CONTEXT_NULL_WHILE_ENABLING_FILE_LOGGING	Context cannot be null while enabling file logging.

9 Appendix

9.1 Business Objects

9.1.1 CW-SDK Configuration Parameters and Preferences

9.1.1.1 Application Wide Configurations

There are static configurations that are required for application and would remain constant for the life of the application.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>Environment</td><td><p><b>Note: Certain checks can be performed only when the application is in specific environment.</b></p><p><b>For example</b></p><ul><li><b>Application installed from authorized source can be performed only when the application is in production environment.</b></li><li><b>CW-SDK performs SafetyNet check asynchronously when the environment is other than DEV and WalletStatus is ACTIVE.</b></li></ul><p>It can further be used for troubleshooting purpose as well. This flag would let SDK know in which environment application is running.</p></td></tr>
<tr><td>LogLevel</td><td><p>Log Level can be used to control CW-SDK logging output to the Android Logcat. Each log level is assigned a priority order. The order in terms of priority from most to least is SILENT &gt; <i>ERROR</i> &gt; WARNING &gt; <i>INFO</i> &gt; <i>DEBUG</i>. And to disable the logging, SILENT can be set. By default, CW-SDK sets log level to ERROR.</p><p><b>Note: SILENT must be kept when application is published to production.</b></p><p>The table shows that log output includes type of logs when current log level is X.</p><table><tr><th>Current Log Level</th><th>Log output</th></tr><tr><td>ERROR</td><td>ERROR</td></tr><tr><td>WARNING</td><td>WARNING + INFO</td></tr><tr><td>INFO</td><td>ERROR + WARNING + INFO</td></tr><tr><td>DEBUG</td><td>DEBUG + INFO + WARNING + ERROR</td></tr><tr><td>SILENT</td><td>No output</td></tr></table></td></tr>
</table>

Attributes	Description
TenantId	Unique tenant Identifier configured during tenant onboarding on Wallet Server.
VelocityCheckRules	Velocity check rules, refer Velocity Check Rules This can be configured only programmatically via *initializeSDK* API.
MDES_PaymentAppProviderId	Unique identifier for the MDES Payment App Provider
MDES_RNS_SenderId	RNS sender Id for MDES
MinimumSUKThreshold	Minimum number of transaction credentials to be kept on wallet instance.
MDES_ServerHostConfiguration	MDES server related configurations. Refer Server Host Configuration for details
Wallet_SERVER_RNS_SenderId	RNS sender Id for Wallet server
Periodic_Sync_Delay_In_Months	The length of the delay in months for the periodic sync request.
Wallet_ServerHostConfiguration	Wallet server related configurations. Refer Server Host Configuration for details
VTS_Signature_PublicKey	VTS Signature Public key is required by URPay to perform integrity check on data received from VTS.
VTS_CvmPriorityConfiguration	VTS CVM Priority configuration is optional. VTS CVM priority can be configured either online PIN priority or CDCVM priority. This configuration determines which priority will be used if the card (via onboarding) supports both CDCVM and online PIN and the terminal requests both or either. If no configuration is set or configuration string is Null or Empty, then online PIN is used as the default. **Note:** • **This configuration String should be the encrypted Base64 value, output by the URPay utility.** • **If input value is not a Base64 string, this will not have any effect on the CVM priority configuration and all card AID priorities will be configured for default value** online PIN priority. • **If** contains **the CVM configuration values that are not supported, all card AID priorities will be configured for default value** online PIN priority. • **If** contains the AID that are not supported, this will not have any effect on the CVM priority configuration and all card AID priorities will be configured for default value online PIN priority. • **The “Supported VTS AID” and “Per AID CVM Priority” should be received during VTS onboarding or from the VISA team.**

Attributes	Description
Google API Key	A unique key which can be used by CW-SDK during Google play services communication for any Google APIs. For example, SafetyNet, Firebase Cloud Messaging, and so on. This key is similar one which is required during tenant on-boarding on Wallet Server. WL-MPA can generate this key by registering the application project on the Google API Console.

9.1.1.2 Server Host configurations

Server Host configuration for Wallet server as well as MDES server.

Attributes	Description
ServerHostURL	Server Host URL for server
CertificateBytes	SSL certificate byte array for server. Please see notes for further details.
ServerPublicKey	A public key for the server
ConnectionTimeOut	Connection timeout during communication with server
ReadTimeOut	Read timeout during communication with server

Note:

There are various ways these configurations can be provided and bundled with the application. Following are few of those:

- **Using gradle build script.**
- **Programmatically via *initializeSDK* API.**
- **In case of *MDES_ServerHostConfiguration*, the *CertificateBytes* should encrypted by URPay utility.**
- **For CertificateBytes in case of MDES, root SSL certificate byte array would be used, which will be encrypted by URPay utility. In case of Wallet Server, intermediate SSL certificate byte array would be used (one above the leaf certificate at the sub-CA level).**

WL-MPA will have to provide these configurations every time it calls the *initializeSDK* method. SDK would keep these configurations in cache and would use it during various business operations when required.

9.1.2 TermsAndCondition

Represents information regarding the terms and condition consent. MPA can call getTermsAndCondition API call to get the latest terms and condition information.

Attributes	Type	Description
termsAndConditionAssetId	String	Unique asset ID of the Terms & Conditions
content	String	Content of Terms & Conditions

9.1.3 DecisioningData

Represents data which are helpful to make decision during digitization process. This object is used during digitize API call providing additional information regarding device.

Attributes	Type	Description
deviceCurrentLocation	String	Latitude and longitude where the device the consumer is attempting to tokenize a card onto is currently located. Optional – This is optional field
deviceIpAddress	String	The current IP address of the device. IPv4 address format of 4 octets separated by "." Ex: 127.0.0.1 Optional – This is optional field
mobileNumberSuffix	String	The last few digits (typically four) of the device's mobile phone number. Optional – This is optional field

9.1.4 CardInfo

Represents Physical Card information which is send as part of digitization request during digitize API call.

Attributes	Type	Description
cardReferenceId	String	Unique reference ID of a Card
accountNumber	String	Account Primary Account Number of the card to be digitized. Length of account number would be between 12 to 19 digits
expiryMonth	String	Month of the Card Expiration date. Format would of 'MM'
expiryYear	String	Year of Card expiration date. Format would be 'YYYY' for Card.
cardHolderName	String	Name of the card holder as mentioned on Card
networkType	Enum	Network type of the Card. Could be MASTERCARD, VISA, MAESTRO etc.
securityCode	String	The CVC2 for the card to be digitized. It should be a 3 digit code. This is optional parameter.

9.1.5 EncryptedCardInfo

Represents Physical Card information in encrypted format, NetworkType and CardReferenceId which is send as part of digitization request during digitize with encrypted API call.

Attributes	Type	Description
cardReferenceId	String	Unique reference ID of a Card
networkType	Enum	Network type of the Card. Could be MASTERCARD, VISA, MAESTRO etc.
encryptedCardInfo	byte[]	Physical card information in bytes data (encrypted)

9.1.6 SyncResult

Represents Synchronization Result Information. This object is received after a successful call to Synchronize API is performed.

Attributes	Type	Description
paymentTokens	ArrayList <Payment Token>	Information of all the updated PaymentTokens. Would return empty list if there are no changes in information. Refer Payment Token structure for details in Payment Token. Those information are provided to CW SDK by Wallet Server.
walletInfo	Wallet Info	Updated wallet information. Optional - Could be null if there is no changes in information. Refer Wallet Info structure for details in Wallet Info.

9.1.7 WalletInfo

Represents information about the wallet. MPA can call getWalletInfo API to get various information of Wallet instance stored locally.

Attributes	Type	Description
walletStatus	Integer	Status of the wallet. INACTIVE, ACTIVE, LOCKED, TERMINATED. In the case of wallet status as TERMINATED, MPA can query TerminateReason.
registeredWithMdes	Boolean	Flag indicating wallet registration status with MDES. Returns TRUE if registered else FALSE
registeredWithVts	Boolean	Flag indicating wallet registration status with VTS. Returns TRUE if registered else FALSE
paymentAppInstanceID	String	PaymentAppInstanceId associated with the Wallet instance. It would be null if wallet is not created.
newAvailableVersionNumber	String	New version number of MPA available for download.

Attributes	Type	Description
		It would be null if new version available information is not present.

9.1.8 PaymentToken

Represents Payment Token Information which is received from CWS on successful call to digitization request for a card.

WL-MPA can call getPaymentTokens API to get a list of PaymentToken stored locally with CW-SDK. MPA can also call getDefaultPaymentToken API also returns the information of default Payment Token, if set.

Attributes	Type	Description
paymentTokenId	String	Unique payment token reference ID
cardReferenceId	String	Unique Card reference ID
accountPanSuffix	String	Last 4 digits of Account PAN
tokenPanSuffix	String	Last 4 digits of D-PAN
networkType	Enum	Network type of the PaymentToken. Could be MASTERCARD or any other card network.
cardProductName	String	Name of card product This is optional
status	Enum	PaymentToken status. It could be either ACTIVE, INACTIVE, SUSPENDED or DEACTIVATED
decision	Enum	The tokenization decision for this digitization request. Must be one of APPROVED, DECLINED, REQUIRE_ADDITIONAL_AUTHENTICATION
activationMethods	List<ActivationMethod>	When additional authentication is required, this is the list of supported activation methods.
asset	CardDisplayAsset	Display attributes of the Card which are required to show the Card on UI. This is optional
isDefault	Boolean	Flag indicating if PaymentToken is set as default.
suspendedBy	List<SuspendedBy >	When PaymentToken is suspended, this list will hold one or more reasons for it.
Digitization Failed Reason	DigitizationFailedReason	Represents reason, i.e. error code & error message when digitization is APPROVED or FAILED in case of digitizeByList API call for a given Payment card.

9.1.9 ActivationMethod

Represents information about all the available Activation Methods. This object is used during a call to requestActivationCode API is performed to request an activation code from Issuer to activate the payment token.

Attributes	Type	Description
methodId	String	Unique identifier assigned to this activation method.
methodType	Enum	Activation method type. Could be one of the following: • TEXT_TO_CARDHOLDER_NUMBER • EMAIL_TO_CARDHOLDER_ADDRESS • CARDHOLDER_TO_CALL_AUTOMATED_NUMBER • CARDHOLDER_TO_CALL_MANNED_NUMBER • CARDHOLDER_TO_VISIT_WEBSITE • CARDHOLDER_TO_USE_MOBILE_APP • ISSUER_TO_CALL_CARDHOLDER_NUMBER This is optional – required in digitize response
methodValue	String	Specifies activation method value. This is optional. Value would depending on type. This is optional – required in digitize response

9.1.10 ActivationResult

It is an Enum representing various results of Activation request. This object is received after a call to Activate API is performed to activate the payment token.

Value	Description
SUCCESS	Activation was successful
INCORRECT_CODE	Activation code was incorrect and rejected by MDES. Retry can be done.
INCORRECT_CODE_RETRIES_EXCEEDED	Activation code was incorrect and rejected. Exceeded the maximum retrial count.
EXPIRED_CODE	Activation code was expired or invalidated.
EXPIRED_SESSION	Digitization session has expired.

9.1.11 CardDisplayAsset

Display assets or attributes of the PaymentToken.

Attributes	Type	Description
url	String	URL of the resource it is located
shortDescription	String	Short description for the card
issuerName	String	Name of the issuer bank for the card
foregroundColor	String	Foreground color of the card.

9.1.12 TransactionInfo

It represents information regarding latest completed transaction.

Attributes	Type	Description
paymentTokenId	String	Unique PaymentToken reference ID
transactionIdentifier	String	Unique identifier for the transaction
amount	Double	The transaction amount.
currencyCode	String - 3-digit ISO 4217 currency code	The transaction currency.
transactionType	String	Type of transaction received from Payment SDK once transaction is completed from POS terminal. Following are the possible transaction types – • PURCHASE • REFUND • UNKNOWN

9.1.13 TransactionDetails

Represents detail of each individual transaction. GetTransactionHistory API returns a list of Transaction Details.

Attributes	Type	Description
paymentTokenId	String	Unique PaymentToken reference ID
transactionIdentifier	String	Unique identifier for the transaction
transactionType	String	Type of Transaction. **Note** Below are the two common transaction type for Mastercard and VISA cards. • PURCHASE • REFUND However, there are some additional transaction type provided by VISA. For example • Cash Withdrawal • ATM Services • ATM Balance Inquiry • AFT • OCT etc. CW-SDK doesn't perform any business based on transaction type received from MDES/VISA. It's responsibility of MPA

Attributes	Type	Description
		to perform their business based on requirement.
amount	Double	The transaction amount.
currencyCode	String - 3-digit ISO 4217 currency code	The transaction currency.
authorizationStatus	Enum	The authorization status of the transaction. Must be one of: AUTHORIZED, DECLINED, CLEARED, and REVERSED.
transactionTimestamp	String	The date/time when the transaction occurred. In ISO 8601 extended format as one of the following: YYYY-MM-DDThh:mm:ss[.sss]Z YYYY-MM-DDThh:mm:ss[.sss]±hh:mm Where [.sss] is optional and can be 1 to 3 digits.
merchantName	String	The merchant (“doing business as”) name.
merchantType	String	The merchant’s type of business or service. Must be a valid Merchant Category Code (MCC).
merchantPostalCode	String	The postal code (for example, zip code in the U.S.) of the merchant.
industryCode	String	The industry code. Conditional – Available only for Visa Cards.
industryCategoryName	String	The industry category name Conditional – Available only for Visa Cards.
industryCategoryCode	String	The industry category code. Conditional – Available only for Visa Cards.
merchantCity	String	The city where the transaction was made. Conditional – Available only for Visa Cards.
atc	String	Application Transaction Counter (ATC). Conditional – Available only for Visa Cards.
transactionScope	String	Scope of the transaction. Conditional – Available only for Visa Cards.

9.1.14 AuthToken

Represents information regarding authentication of application. It contains value Identity token only.

Attributes	Type	Description
identityToken	String	Identity Token received from ID&V
dataToken	String	Data Token

9.1.15 AuthenticationState

Represents information regarding authentication state of a user including authentication status and last authentication date.

Attributes	Type	Description
isAuthenticated	Boolean	Flag indicating is user authenticated for Tap N Pay transaction.
lastAuthenticationTime	Date	DateTime when user was last authenticated.

9.1.16 VelocityCheckRules

Velocity check rules are specified in Specification document. Please refer Customizable_NFC_Wallet_Functional_Specification under section References.

CW SDK will treat the session duration as infinite when the given session duration values are not provided within the configuration at a time to CW SDK initialization process.

Attributes	Type	Description
LvtAmountThresholdNCurrencyCode	Map	The map of country specific LVT Threshold value and Currency code managed at the WL-MPA level for phase. 3 sets of (LVT Threshold, Currency code) should be managed (see “LVT Threshold Value / Currency Code” in FS)
LvtCounterThreshold	Integer	LVT transaction counter limit
HvtCounterThreshold	Integer	HVT transaction counter limit
LvtSessionDuration	Long	LVT transaction session duration This value is in second. For example, 5 will set session duration to 5 seconds.
HvtSessionDuration	Long	HVT transaction session duration This value is in second. For example, 5 will set session duration to 5 seconds.

Attributes	Type	Description
AccumulatedAmtThreshold	Double	Total Transaction Accumulated amount limit

9.1.17 Operation

Following table specifies the list of Operation used for DataToken (L2) generation.

Operation	Description
CREATE_WALLET	Use to generate DataToken for createWallet API
REGISTER_WITH_MDES	Use to generate DataToken for registerWithMdes API
REGISTER_WITH_VTS	Use to generate DataToken for registerWithVts API
DIGITIZE	Use to generate DataToken for digitize API

9.1.18 OperationParams

Following table specifies the list of operation params used for DataToken (L2) generation.

<table>
<tr><th>OperationParams</th><th>Description</th></tr>
<tr><td>CreateWalletOperationParams</td><td>List of Params required for DataToken generation for CREATE_WALLET operation.
<table>
<tr><th>Attributes</th><th>Type</th><th>Description</th></tr>
<tr><td>walletServerRnsRegistrationId</td><td>String</td><td>This is the Wallet Server Registration Id received by calling the getToken API while using Wallet Server’s sender Id as authorizedEntity</td></tr>
<tr><td>cesRnsRegistrationId</td><td>String</td><td>This is the CES Registration Id received by calling the getToken API while using CES sender Id as authorizedEntity.</td></tr>
</table></td></tr>
<tr><td>RegisterWithMdesOperationParams</td><td>List of Params required for DataToken generation for REGISTER_WITH_MDES operation.
<table>
<tr><th>Attributes</th><th>Type</th><th>Description</th></tr>
<tr><td>mdesRnsRegistrationId</td><td></td><td>This is the MDES Registration Id</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>OperationParams</th><th>Description</th></tr>
<tr><td></td><td><table><tr><td></td><td></td><td>received by calling the getToken API while using MDES sender Id as authorizedEntity.</td></tr></table></td></tr>
<tr><td>RegisterWithVtsOperationParams</td><td>No additional parameter required for REGISTER_WITH_VTS operation.</td></tr>
<tr><td>DigitizeOperationParams</td><td>List of Params required for DataToken generation for DIGITIZE operation.<table><tr><th>Attributes</th><th>Type</th><th>Description</th></tr><tr><td>cardInfo</td><td>CardInfo</td><td>Refer CardInfo in Appendix</td></tr><tr><td>decisioningData</td><td>DecisioningData</td><td>Refer DecisioningData in Appendix</td></tr></table></td></tr>
</table>

Note:

To check how to obtain the registration id for multiple firebase project, Please refer Firebase Cloud Messaging Configuration section in the integration guide.

9.1.19 TransactionError

Represent error occurred during the payment transaction.

Attributes	Type	Description
errorCode	String	Error code

9.1.20 SuspendedBy

Represents reasons by which a PaymentToken is suspended.

Attributes	Description
ISSUER	PaymentToken is suspended by Issuers.
PAYMENT_APP_PROVIDER	PaymentToken is suspended by Payment Application Provider.
MOBILE_PIN_LOCKED	PaymentToken is suspended due to Mobile PIN is being locked
CARDHOLDER	PaymentToken is suspended by card holders.

9.1.21 EnvironmentCheck

Represents constants indicating environment checks operations to be performed during performEnvironmentalChecks API Call. Same will be returned in response map.

Attributes	Description
AUTHORIZED_SOURCE	If set in parameter, API will check if application is installed from authorized source or not.
ROOTED_DEVICE	If set in parameter, API will check if device is rooted or not
NFC_SUPPORT	If set in parameter, API will check if device have NFC feature or not
NFC_ENABLED	If set in parameter, API will check if NFC is enabled or not on the device.
SECURE_UNLOCK_MECHANISM_ENABLED	If set in parameter, API will check if device secure unlock is enabled or not.

9.1.22 TransactionContext

Represents contextual information like Terminal type version for an on-going transaction.

Attributes	Type	Description
terminalType	Enum	Terminal versions like MASTERCARD_TERMINAL_V2, MASTERCARD_TERMINAL_V3 and UNKNOWN_TERMINAL.

9.1.23 RecoveryStatus

Represents status of the recovery process initiated by startRecovery API call. MPA can also call getRecoveryStatus API to check the status of the recovery process at any time.

Attributes	Description
NONE	When recovery process is not required
REQUIRED	Recovery process is required to restore previous wallet state. Call startRecovery API.
PARTIAL	Recovery process is partial and there are still some information needs to be recovered. Keep calling startRecovery to complete the recovery process.
COMPLETED	Recovery process is completed.

9.1.24 DigitizationFailedReason

Represents digitization failed reason of a given PaymentToken in the form of error code and error message when digitizeByList API is used to digitize multiple cards.

Attributes	Type	Description
errorCode	String	Error code
errorMessage	String	Error message
errorReason	String	Error reason

Note:

When value of the error code is “PROVISION_FAILED”, then MPA should delete that payment token by calling deletePaymentToken API.

9.1.25 TerminateReason

Represents reason of wallet termination. MPA can leverage retrieveTerminateReason API from WalletLifeCyclePreferenceHelper utility class to know the wallet termination reason in detail.

Attributes	Type	Description
NO_REASON_FOUND	enum	When there is no reason stored
APPLICATION_INITIATED	enum	Wallet Termination was initiated by MPA using terminateWallet API
CWS_INITIATED	enum	Wallet termination was initiated by Wallet Server. It may be due to a request from Issuer or Customer Service Portal.
DEVICE_ROOT_VERIFICATION_FAILED	enum	Wallet was terminated due to the device root verification check is failed
SAFETYNET_VERIFICATION_FAILED	enum	Wallet was terminated due to Google SafetyNet verification is failed
INITIALIZATION_FAILED	enum	Wallet was terminated and reset due to native initialization error from URPay during initializeSDK call

9.1.26 LoggingProperties

Represents configurations for file based logging. Which defines file name, parent directory, maximum number of files to be generated, maximum size of the file.

Attributes	Type	Description
FILE_NAME	String	Name of the file which will be generated while file based logging
MAX_FILE_SIZE_KB	Long	Maximum size of the log file
MAXMIUM_NUMBER_OF_FILE	Long	Maximum number of files to be generated for logs

Attributes	Type	Description
PARENT_DIR	String	Name of the parent directory of log files

9.1.27 Un-Encrypted Card Information Sample

Below is the sample JSON format of un-encrypted card information which should be encrypted by the Santander Server. The WL-MPA must receive the encrypted Card Information from the Santander Server.

```
{
"accountNumber":"1234123412341234",
"cardReferenceId":"7f9c08af3da083281f19229109e7cbb962ff74c28a15477463a0bfe5f26f5747",
"cardholderName":"Card Holder Name",
"expiryMonth":"12",
"expiryYear":"2025",
"networkType":"MASTERCARD",
"securityCode":"123"
}
```

9.2 Error Descriptions

Description for all the error codes raised by CW-SDK.

Error Code	**Description**
INITIALIZATION_FAILED	If Initialization fails due to failure of initialization of Payment SDK or for some other reason.
INVALID_CONFIGURATIONS	If the mandatory configurations are not passed or any invalid configurations are passed.
SDK_NOT_INITIALIZED	If CWSDK is not initialized. **Note:** **CWSDK can be initialized using initializeSDK method.**
SYSTEM_ERROR	Any technical exception/error while performing all the checks.
NO_DATA_CONNECTIVITY	No Data connectivity available.
DEVICE_NOT_SUPPORTED	Device not supported Meaning, current device is either black listed or rooted. In this case, WL-MPA should not allow the application to further invoke other APIs. WL-MPA can use Terminate Wallet API to delete the wallet data if needed.
OS_NOT_SUPPORTED	OS is not supported Meaning, current Operating System of the device is not supported. In this case, WL-MPA should not allow the

Error Code	Description
	application to further invoke other APIs. WL-MPA can use Terminate Wallet API to delete the wallet data if needed.
COMMUNICATION_ERROR	Error while communicating with remote server
INTERNAL_SYSTEM_ERROR	Some unknown internal service failure
INVALID_WALLET_STATE	The Wallet is in an invalid status for the requested operation.
INVALID_INPUT	Required input arguments are invalid.
AUTH_TOKEN_AUTHORIZATION_FAILED	ID&V token authorization failed.
INVALID_WALLET_PIN_FORMAT	Format of the Wallet PIN entered is invalid
MDES_REGISTRATION_FAILED	Registration with MDES fails.
TNC_REQUIRED	Terms and Conditions is required to digitize the card.
TNC_ACCEPTANCE_FAILED	Terms and Conditions acceptance failed on WS or MDES/VTS.
CARD_ALREADY_DIGITIZED	Card was already digitized previously.
TNC_NOT_AVAILABLE	Terms and Conditions is not available on Wallet Server
ACTIVATION_ERROR	It could be due to other reason like session expired.
ACTIVATION_PERIOD_EXPIRED	Activation period is expired. Refer Notes below
UNKNOWN_PAYMENT_TOKEN	PaymentToken is unknown.
INVALID_PAYMENT_TOKEN_STATUS	The token is in an invalid status for the requested operation.
INVALID_WALLET_PIN	Invalid PIN
NFC_DISABLED	NFC is disabled.
APP_NOT_DEFAULT_FOR_PAYMENT	Application is not selected as default for payment in tap & pay settings
PAYMENT_TOKEN_EXHAUSTED	No more transaction credentials available for making payment.
ERROR_FETCHING_TRANSACTION_HISTORY	Error while fetching transaction history information from Server.
NO_CARD_ACTIVE_FOR_PAYMENT	The user tried to pay, but there was no default payment card found.
FIRST_TAP_TIMEOUT	The payment time out if transaction is initiated by initiateTransaction.
DEVICE_LEFT_NFC_FIELD	The device left the NFC field while payment transaction was in progress.
TRANSACTION_CONTEXT_CONFLICT	Second Tap was performed by user but terminal had changed the amount or currency since the First Tap.
UNKNOWN_ERROR	Transaction failed due to reason not known or condition of use is not satisfied.

Error Code	Description
MISSING_EXPIRY_DATE	The expiry date is required for this product but was missing.
PAN_INELIGIBLE_FOR_DEVICE	The PAN is not allowed to be provisioned to the device because of Issuer rules.
PAN_PROVISIONING_COUNT_EXCEEDED	The PAN has already been provisioned to the maximum number of devices.
DEVICE_INELIGIBLE	The device is not supported for use with MDES/VTS.
INVALID_PAN	The PAN format is not valid, or other data associated with the PAN was incorrect or entered incorrectly.
PAN_INELIGIBLE	The PAN is not in an approved account range for MDES/VTS.
PAYMENT_TOKEN_EXHAUSTED	No more transaction credentials available for making payment.
INVALID_ACTIVATION_METHOD	The activation method could not be found.
PAYMENT_TOKEN_SUSPENDED_FOR_TRANSACTION_HISTORY	The token is in suspended status for fetching transaction history.
PAYMENT_TOKEN_INACTIVE_FOR_TRANSACTION_HISTORY	The token is in inactive status for fetching transaction history.
PROVISION_FAILED	The provisioning of PAN to the device has failed. Please delete this card. **Note:** When this exception received, its responsibility of MPA to delete that payment token by calling deletePaymentToken API.
RESUME_PAYMENT_TOKEN_FAILED	Resume of Payment Token failed on WS due to some specific reason. For example, Payment Token suspended by Issuer cannot be resumed using this API.
TRANSACTION_APPLICATION_DECLINED	Transaction is aborted due to various reasons like consent not given, wallet state is not ACTIVE and device is rooted and other reasons.
WALLET_ALREADY_REGISTERED	Wallet is already registered with MDES.
WALLET_NOT_REGISTERED_WITH_TSP	Wallet is not registered with MDES/VTS.
CARD_OPERATION_BLOCKED	Further operations for this card are no longer allowed. Please contact your bank to resolve this issue. Note: This exception will only occur in the case of Visa.
AUTH_TOKEN_CREDENTIAL_MISMATCH	ID&V credentials passed during current API call is not same as ID&V credentials used during wallet creation. It should be treated as unauthorized access of that API call. Meaning, it may happen that wallet creation is done by different user whereas during other API calls,

Error Code	Description
	AuthToken input parameter is having credentials of a different user.
INVALID_WORKFLOW	There is any error in workflow. The card activation is called before requesting for activation code, SUSPEND or RESUME called on INACTIVE Token, etc.
FILE_LOGGING_NOT_ALLOWED_IN_RELEASE_MODE	Enabling file based logging is not allowed in release mode of CW-SDK. This is to prevent accidental usage of file based logging when WL-MPA is published over Google Play Store. File based logging must only be used for CW-SDK troubleshooting prior to publishing it for end users.
CONTEXT_NULL_WHILE_ENABLING_FILE_LOGGING	Context cannot be null while enabling file logging. This is a mandatory parameter.
RESET_REQUIRED	This error would occur in cases when URPay SDK data has wiped out i.e. Wallet PIN, PaymentToken etc. In this scenario MPA must call reset API to reset the wallet.

9.3 Android Permissions

Android App needs explicit permissions and consent from user to perform to perform certain operations.

Few of the method within the CW-SDK will require some permissions. Refer the **Android Permissions_Worksheet_v1.0** excel file for information regarding the permissions required for CW-SDK.

Perform below steps to access the **Android Permissions_Worksheet_v1.0** file:

- Unzip **FLAME_CWSDK_API_Specification_VX.X.X.zip** folder -> Open **Android Permission** folder -> Refer **Android Permissions_Worksheet_v1.0.xlsx** file

The Play Store shows the user a list of device services that application will use (for example NFC/Camera and other device services) and prompts the User to consent and grant permission to the application to use those services and initiate the download.