Android

This document contains the Programmable Wallets SDK reference for Android development.Suggest Edits

The User Wallet provides an App SDK in web, iOS, and Android for the customer to integrate. The App SDK secures the process when users input their secret data, whether as PIN or security answer, and the SDK app encrypts the request body by the secret key.

Due to the security requirement for Circle to control the end-to-end process, the SDK also exposes functionality for the customer to customize the description and layout:

- UI Title and Subtitle Customization:
- Modify the title and subtitle to reflect your brand identity or provide specific instructions.
- Custom PIN Code Input Layout:
- Adjust the layout and styling of the PIN code input field to align with your application's design guidelines.
- Question List Configuration:
- Set the list of security questions displayed in the User Wallet UI, allowing users to choose from a predefined set.
- SDK Initialization:
- Initialize the Web SDK by setting the endpoint server, ensuring seamless communication between your application and our services.
- Predefined Error Messages:
- Customize the error messages displayed to users, providing a more personalized experience and guidance.
- ChallengeID Acceptance and Operation Retrieval:
- Easily accept the challengeID and retrieve any relevant operations within the SDK.

Tip:

To use the SDK in the most flexible way, combine this SDK reference with theAndroid SDK UI Customization API article.

SDK Download

Download the Android SDKhere .

Android SDK

Requirements

Our Android SDK supports Android API level 21+. Earlier versions are not supported.

Recommend using the latest version of Android Studio.

WalletSdk

object WalletSdk

Public Methods @Throws(Throwable::class) Unit init(context: Context?, config: WalletSdk.Configuration?)

Configure the Circle endpoint for SDK to connect, throw Throwable if the parameters are null or endpoint and appId's format are invalid. Unit setLayoutProvider(provider: LayoutProvider?)

Set a LayoutProvider derived class for customization, e.g. error code message, font and color. Ignore when the parameter is null. Unit setViewSetterProvider(provider: ViewSetterProvider?)

Set a ViewSetterProvider derived class for image customization. @Throws(Throwable::class) Unit setSecurityQuestions(questions: Array?)

Set the security question list, throw Throwable when the SecurityQuestion array is empty or contains any question whose title length is not 2~512. Unit execute(activity: Activity?, userToken: String?, encryptionKey: String?, challengeId: Array?, callback: Callback?)

Execute the operations by challengeId and call the callback after sending the request to Circle endpoint. Ignore when callback is null. Callback will receive onError() when parameters are null or invalid. Unit addEventListener(listener: EventListener?)

Register an EventListener for the app to handle events, e.g. forgot PIN. Ignore when the parameter is null. Unit

removeEventListener(listener: EventListener?)

Unregister an EventListener. Ignore when the parameter is null. Unit moveTaskToFront(context: Context?)

Bring the SDK UI to foreground. If the app launches another Activity in onEven() and onError() and makes the SDK UI in background, use this API to go back to the SDK UI. Ignore when the parameter is null. Unit setBiometricsPin(activity: Activity?, userToken: String?, encryptionKey: String?, callback: Callback?)

Setup BiometricsForPin and call the callback after operation. Ignore when callback is null. Callback will receive onError() when parameters are null or invalid. Unit setCustomUserAgent(userAgent: String?)

Set custom user agent value. Ignore when the parameter is null. String sdkVersion()

Get SDK version.

EventListener

EventListener interface that receives events when an event is triggered.

interface EventListener

Public Methods abstract Unit onEvent(event: ExecuteEvent)

Called when the event triggered.

ExecuteEvent

ExecuteEvent, see WalletSdk.addEventListener().

enum ExecuteEvent

Enum values forgotPin

WalletSdk.Configuration

SDK Configuration for WalletSdk init.

data class Configuration

Constructors constructor(endPoint: String?, appId: String?)

Init with Circle endpoint. SDK will connect to this endpoint. constructor(endpoint: String?, appId: String?, settingsManagement: SettingsManagement?)

Init with Circle endpoint. SDK will connect to this endpoint. The SettingsManagement can bring extra setting flags to use.

SettingsManagement

SettingsManagement use to bring extra setting flags to Configuration that would be used when initial WalletSdk.

data class SettingsManagement

Constructors constructor(isEnableBiometricsPin: Boolean = false) Properties Boolean isEnableBiometricsPin

Flag that decides whether to use biometrics to protect PIN or not.

LayoutProvider

LayoutProvider helps perform customization during runtime.

open class LayoutProvider

Public Methods open TextConfig? getTextConfig(key: String)

Define strings with specific configurations for general string customization. Returned TextConfig will replace strings.xml, colors.xml, and styles.xml values.

All keys are listed inC Index Table . open Array? getIconTextConfigs(key: Resource.IconTextsKey)

Define icon and string sets with specific configurations for icon text list item customization.

All keys are listed inB Index Table . open Array? getTextConfigs(key: Resource.TextsKey)

Define strings with specific configurations for special text customization.

All keys are listed inA Index Table . open String? getErrorString(code: ApiError.ErrorCode)

Define the error description.

All error codes are listed in ApiError.ErrorCode. open String? getDateFormat()

Get display date format, e.g. the answer of a security question in which inputType is datePicker.

Only those 3 strings are valid values: 1. “YYYY-MM-DD”, 2. “DD/MM/YYYY” 3. “MM/DD/YYYY”

If it returns an invalid value, the default value would be “YYYY-MM-DD”.

All supported formats are listed Resource.DateFormat. open Boolean isDebugging()

true : default value, check returned value, and print warn level log

false : skip checking and turn off the log.

Resource.DateFormat

interface DateFormat

Constants String YYYYMMDD_HYPHEN = "YYYY-MM-DD"

DDMMYYYY_SLASH = "DD/MM/YYYY"

MMDDYYYY_SLASH = "MM/DD/YYYY"

Available values of LayoutProvider.getDateFormat().

Resource.Key

interface Key

Static Fields String circlepw_show_pin = "circlepw_show_pin"

circlepw_hide_pin = "circlepw_hide_pin"

... SeeC Index Table .

TextConfig

Data class for text customization.

data class TextConfig

Constructors constructor(text: String?, gradientColors: IntArray?, font: Typeface?) constructor(text: String?, textColor: Int, font: Typeface?) constructor(text: String?) Properties String? text

Text to display. @ColorInt IntArray? gradientColors

Array of Gradient text color. Only used by

TextsKey.enterPinCodeHeadline ,

TextsKey.securityIntroHeadline ,

TextsKey.newPinCodeHeadline @ColorInt IntArray? textColor

Text color. Typeface? font

Font.

IconTextConfig

Data class for icon text list item customization.

data class IconTextConfig

Constructors constructor(setter: IImageViewSetter?, textConfig: TextConfig?) Properties IImageViewSetter? setter

The ImageView setter for image customization. TextConfig? textConfig

Text config for text customization.

Resource.TextsKey

enum TextsKey

Enum Values securityQuestionHeaders, securitySummaryQuestionHeaders, enterPinCodeHeadline, securityIntroHeadline, newPinCodeHeadline, securityIntroLink, recoverPinCodeHeadline

SeeA Index Table .

Resource.IconTextsKey

enum IconTextsKey

Enum Constants securityConfirmationItems

SeeB Index Table .

IImageViewSetter

The ImageView setter interface for image customization.

interface IImageViewSetter

Public Methods abstract Unit apply(iv:ImageView?) Called when the ImageView needs to be set.

LocalImageSetter

The implemented ImageView setter for image customization with local image.

class LocalImageSetter: IImageViewSetter

Constructors constructor(drawableId: Int) Properties @DrawableRes Int drawableId

The resource ID of drawable. Public Methods Unit apply(ivImageView) Set drawable to the imageView with drawableId.

RemoteImageSetter

The implemented ImageView setter for image customization with a remote image.

class RemoteImageSetter: IImageViewSetter

Constructors constructor(drawableId: Int) constructor(drawableId: Int, url: String?) Properties @DrawableRes Int drawableId

The resource ID of drawable. String url

Image URL. Public Methods Unit apply(iv:ImageView) Set a remote image from the URL to the ImageView. Use the drawable as the placeholder.

IToolbarSetter

The Toolbar setter interface for image customization.

interface IToolbarSetter

Public Methods abstract Unit apply(toolbar:Toolbar?) Called when the Toolbar needs to be set.

LocalToolbarImageSetter

class LocalToolbarImageSetter: IToolbarSetter

Constructors constructor(drawableId: Int) Properties @DrawableRes Int drawableId

The resource ID of drawable. Public Methods Unit apply(toolbar:Toolbar?) Set drawable as navigation icon to the Toolbar.

RemoteToolbarImageSetter

The implemented Toolbar setter for image customization with a remote image.

class RemoteToolbarImageSetter: IToolbarSetter

Constructors constructor(drawableId: Int) constructor(drawableId: Int, url: String?) Properties @DrawableRes Int drawableId

The resource ID of drawable. String? url

Image URL. Public Methods Unit apply(toolbar:Toolbar?) Set a remote image from the URL to the toolbar. Use drawable as the placeholder.

ViewSetterProvider

ViewSetterProvider supports performing image customization during runtime.

open class ViewSetterProvider extends Object

Public Methods open IImageViewSetter? getImageSetter(type: Resource.Icon)

Return implemented IImageViewSetter for performing general image customization. e.g. LocalImageSetter, RemoteImageSetter

All keys are listed inD Index Table . open IToolbarSetter? getToolbarImageSetter(type: Resource.ToolbarIcon)

Return implemented IToolbarSetter for performing Toolbar image customization. e.g. LocalToolbarImageSetter, RemoteToolbarImageSetter

All keys are listed inE Index Table .

Resource.Icon

enum Icon

Enum Values securityIntroMain, selectCheckMark, dropdownArrow, errorInfo, securityConfirmMain, biometricsAllowMain, showPin, hidePin, alertWindowIcon

D Index Table .

Resource.ToolbarIcon

enum ToolbarIcon

Enum Values close, back

E Index Table .

SecurityQuestion

Data class for security questions customization. See WalletSdk.setSecurityQuestions().

data class SecurityQuestion

Constructors constructor(title: String) constructor(title: String, inputType: SecurityQuestion.InputType) Properties String title

The question string. SecurityQuestion.InputType inputType

The input type of the question. Support text input and timePicker.

SecurityQuestion.InputType

public enum InputType

Enum Values text, datePicker

Callback

A generic callback interface for SDK API calls

interface Callback

Public Methods abstract Boolean onError(error: Throwable)

The callback is triggered when a failure occurs in operation or is canceled by the user. Return true - The app will handle the following step, SDK will keep the Activity. Return false - The app won’t handle the following step, SDK will finish the Activity. abstract Init onResult(R result)

R - Type of result such as ExecuteResult Callback when the operation is executed successfully. Finish the Activity after this callback is triggered. abstract Boolean onWarning(ExecuteWarning warning, R result)

The callback is triggered when operation executed with warning. R - Type of result such as ExecuteResult Return true - App will handle the following step, SDK will keep the Activity. Return false - App won’t handle the following step, SDK will finish the Activity.

ExecuteWarning

enum ExcuteWarning

Properties Int warningType

Warning type. String warningString

Description of the warning type. Enum Values biometricsUserSkip(155711, "User skipped the setting of using biometrics to protect PIN this time."), biometricsUserDisableForPin(155712, "User disabled the function of using biometrics to protect PIN."), biometricsUserLockout(155713, "Too many attempts. Try again later."), biometricsUserLockoutPermanent(155714, "Too many attempts. Biometrics sensor disabled."), biometricsUserNotAllowPermission(155715, "User didn't grant the permission to use biometrics"), //IOS ONLY biometricsInternalError(155716, "Biometrics internal error - %s");

ExecuteResult

data class ExecuteResult

Constructors constructor(resultType: ExecuteResultType, status: ExecuteResultStatus, data: ExecuteResultData = ExecuteResultData()) Properties ExecuteResultType resultType

The type of the operation that the challenge represents. The possible values are : UNKNOWN(-1), SET_PIN(0), RESTORE_PIN(1), CHANGE_PIN(2), SET_SECURITY_QUESTIONS(3), CREATE_WALLET(4), CREATE_TRANSACTION(5), ACCELERATE_TRANSACTION(6), CANCEL_TRANSACTION(7), CONTRACT_EXECUTION(8), SIGN_MESSAGE(9), SIGN_TYPEDDATA(10), SET_BIOMETRICS_PIN(1000) ExecuteResultStatus status

The status of the execution. The possible values are : UNKNOWN(-1) PENDING(0) IN_PROGRESS(1) COMPLETE(2) FAILED(3) EXPIRED(4) ExecuteResultData data

The data of the execution.

ExecuteResultType

enum ExecuteResultType

Properties Int value

The int value of the enum. Enum Values UNKNOWN(-1), SET_PIN(0), RESTORE_PIN(1), CHANGE_PIN(2), SET_SECURITY_QUESTIONS(3), CREATE_WALLET(4), CREATE_TRANSACTION(5), ACCELERATE_TRANSACTION(6), CANCEL_TRANSACTION(7), CONTRACT_EXECUTION(8), SIGN_MESSAGE(9), SIGN_TYPEDDATA(10), SET_BIOMETRICS_PIN(1000)

ExecuteResultStatus

enum ExecuteResultStatus

Properties Int value

The int value of the enum. Enum Constants UNKNOWN (-1) PENDING(0) IN_PROGRESS(1) COMPLETED(2) FAILED(3) EXPIRED(4)

ExecuteResultData

data class ExecuteResultData

Constructors constructor(signature: String? = null) Properties String? signature

The signature for SIGN_MESSAGE and SIGN_TYPEDDATA.

APIError

Error class for PW SDK

abstract class ApiError:Throwable

Properties abstract ApiError.ErrorCode code

Error code. open String message

Get human-readable error message

ApiError.ErrorCode

public enum ErrorCode

Properties Int value

The int value of the enum. Enum Values unknown(-1), success(0), apiParameterMissing(1), apiParameterInvalid(2), forbidden(3), unauthorized(4), retry(9), customerSuspended(10), pending(11), invalidSession(12), invalidPartnerId(13), invalidMessage(14), invalidPhone(15), walletIdNotFound(156001), tokenIdNotFound(156002), transactionIdNotFound(156003), entityCredentialNotFound(156004), walletSetIdNotFound(156005), userAlreadyExisted(155101), userNotFound(155102), userTokenNotFound(155103), userTokenExpired(155104), invalidUserToken(155105), userWasInitialized(155106), userHasSetPin(155107), userHasSetSecurityQuestion(155108), userWasDisabled(155109), userDoesNotSetPinYet(155110), userDoesNotSetSecurityQuestionYet(155111), incorrectUserPin(155112), incorrectDeviceId(155113), incorrectAppId(155114), incorrectSecurityAnswers(155115), invalidChallengeId(155116), invalidApproveContent(155117), invalidEncryptionKey(155118), userPinLocked(155119), securityAnswersLocked(155120), walletIsFrozen(155501), maxWalletLimitReached(155502), walletSetIdMutuallyExclusive(155503), metadataUnmatched(155504), userCanceled(155701), launchUiFailed(155702), pinCodeNotMatched(155703), insecurePinCode(155704), hintsMatchAnswers(155705); biometricsSettingNotEnabled(155708) deviceNotSupportBiometrics(155709) biometricsKeyPermanentlyInvalidated(155710) biometricsUserSkip(155711) biometricsUserDisableForPin(155712) biometricsUserLockout(155713) biometricsUserLockoutPermanent(155714) biometricsUserNotAllowPermission(155715) biometricsInternalError(155716)

Static Customized String

res/values/strings.xml

XML Continue Next Question:

Static Customized UI Layout

res/values/color.xml

XML

#1A1A1A #3D3D3D #136FD8 #B3136FD8

res/values/styles.xml

XML

res/values/dimens.xml

XML 10dp 24dp 48dp

Sample Code

Java // Set endpoint and app ID try { WalletSdk.init(getApplicationContext(), new WalletSdk.Configuration(endpoint, appId)); } catch (Throwable e){ e.printStackTrace(); return; }

// setViewSetterProvider WalletSdk.setViewSetterProvider(new MyViewSetterProvider()); // setLayoutProvider WalletSdk.setLayoutProvider(new MyLayoutProvider(getApplicationContext())); // Regester EventListener WalletSdk.addEventListener(new EventListener() { @Override public void onEvent(ExecuteEvent event) { switch (event){

case forgotPin: startActivity(forgotPinIntent);
break; } } });

// setSecurityQuestion SecurityQuestion[] questions = new SecurityQuestion[]{ new SecurityQuestion("What is your father's middle name?"), new SecurityQuestion("What is your favorite sports team?"), new SecurityQuestion("What is your mother's maiden name?"), new SecurityQuestion("What is the name of your first pet?"), new SecurityQuestion("What is the name of the city you were born in?"), new SecurityQuestion("What is the name of the first street you lived on?"), new SecurityQuestion("When is your father's birthday?", SecurityQuestion.InputType.datePicker)};
WalletSdk.setSecurityQuestions(questions); Java // There are two ways to use Configuration. Below is the way with SettingsManagement. // We can use SettingsManagement to bring the setting flag. SettingsManagement provide the setting flag // EnableBiometricsPin to enable/disable the function "Use biometrics to protect PIN" currently. // But if we do not use the settingsManagement, the default value of EnableBiometricsPin is false. try { SettingsManagement settingsManagement = new SettingsManagement(); settingsManagement.setEnableBiometricsPin(true); //Set "true" to enable, "false" to disable WalletSdk.init(getApplicationContext(), new WalletSdk.Configuration(endpoint, appId, settingsManagement)); } catch (Throwable e){ e.printStackTrace(); return; } Java // Execute WalletSdk.execute(getActivity(), userToken, encryptionKey, new String[]{challengeId}, new Callback() { @Override public boolean onError(Throwable error) { if(error instanceof ApiError){ ApiError apiError = (ApiError) error; if(apiError.getCode() == ApiError.ErrorCode.userCanceled){ // App won't handle next step, SDK finishes the Activity. return false;
} startActivity(errorUiIntent);
// App will handle next step, SDK keeps the Activity. return true; }

```
        }


        @Override
        public void onResult(ExecuteResult result) {
              // success
        }


        @Override
        public boolean onWarning(ExecuteWarning warning, ExecuteResult executeResult) {
        // App won't handle next step, SDK finishes the Activity.
              return false;
        }
```

}); Java // Extend LayoutProvider and override class MyLayoutProvider extends LayoutProvider { Context context; Typeface typeface; int color; public MyLayoutProvider(Context context) { this.context = context; }

```
  @Override
  public TextConfig getTextConfig(String key) {
     switch (key){
        case Resource.Key.circlepw_security_intros_description:
           typeface = ResourcesCompat.getFont(context, R.font.en);
           color = Color.parseColor("#105AAB");
           return new TextConfig("customized description", color, typeface);
     }
     return super.getTextConfig(key);
  }
```

@Override public IconTextConfig[] getIconTextConfigs(Resource.IconTextsKey key) { switch (key){ case securityConfirmationItems: // SQ confirmation introduction items return new IconTextConfig[]{ new IconTextConfig(new LocalImageSetter(R.drawable.intro_item0_icon), new TextConfig("This is the only way to recover my account access. ")), new IconTextConfig(new RemoteImageSetter(R.drawable.error, "https://path/icon2.svg"), new TextConfig("Circle won't store my answers so it's my responsibility to remember them.")), new IconTextConfig(new RemoteImageSetter(R.drawable.error, "https://path/icon3.svg"), new TextConfig("I will lose access to my wallet and my digital assets if I forget my answers. ")) }; } return super.getIconTextConfigs(key); } @Override public TextConfig[] getTextConfigs(Resource.TextsKey key) { switch (key){ case securityQuestionHeaders: return new TextConfig[]{ new TextConfig("Choose your 1st question"), new TextConfig("Choose your 2nd question") }; case securitySummaryQuestionHeaders: return new TextConfig[]{ new TextConfig("1st Question"), new TextConfig("2nd Question") }; case newPinCodeHeadline: case enterPinCodeHeadline: return new TextConfig[]{ new TextConfig("Enter your "), new TextConfig("PIN", getHeadingColor(), null), }; case securityIntroHeadline: return new TextConfig[]{ new TextConfig("Set up your "), new TextConfig("Recovery Method", getHeadingColor(), null), }; } return super.getTextConfigs(key); }

```
  private int getHeadingColor(){
       return Color.parseColor("#0073C3");
  }

  @Override
  public String getErrorString(ApiError.ErrorCode code) {
     switch (code){
        case userCanceled:
           return context.getString(R.string.user_canceled);
     }
```

```
        return super.getErrorString(code);
    }

    @Override
    public String getDateFormat(){
        return Resource.DateFormat.MMDDYYYY_SLASH; // "MM/DD/YYYY"
    }

    @Override
    public boolean isDebugging(){
        return BuildConfig.DEBUG;
    }

}
```

Java // Extend ViewSetterProvider and override class MyViewSetterProvider extends ViewSetterProvider{

```
    @Override
    public IToolbarSetter getToolbarImageSetter(Resource.ToolbarIcon type) {
        switch (type){
            case back:
                return new LocalToolbarImageSetter(R.drawable.ic_back);
            case close:
                return new RemoteToolbarImageSetter(R.drawable.error, "https://path/close.svg");
        }
        return super.getToolbarImageSetter(type);
    }
@Override
    public IImageViewSetter getImageSetter(Resource.Icon type) {
        switch (type){
            case securityIntroMain:
                return new LocalImageSetter(R.drawable.main);
            case selectCheckMark:
                return new RemoteImageSetter(R.drawable.error, "https://path/check.svg");
            case dropdownArrow:
                return new RemoteImageSetter(R.drawable.error, "https://path/arrow.svg");
            case errorInfo:
                return new RemoteImageSetter(R.drawable.error, "https://path/errorinfo.svg");
            case securityConfirmMain:
                return new RemoteImageSetter(R.drawable.error, "https://path/main2.svg");
        }
        return super.getImageSetter(type);
    }
}
```

Java // setBiometricsPin WalletSdk.setBiometricsPin(this, userToken, encryptionKey, new Callback() { @Override public boolean onError(Throwable error) { error.printStackTrace(); if (error instanceof ApiError) { ApiError apiError = (ApiError) error; switch (apiError.getCode()) { case incorrectUserPin: case userPinLocked: case securityAnswersLocked: case incorrectSecurityAnswers: case pinCodeNotMatched: case insecurePinCode: return true; // App will handle next step, SDK will keep the Activity. } } return false; // App won't handle next step, SDK will finish the Activity. }

```
        @Override
        public void onResult(ExecuteResult executeResult) {
            //success
        }

        @Override
        public boolean onWarning(ExecuteWarning warning, ExecuteResult executeResult) {
            return false; // App won't handle next step, SDK will finish the Activity.
        }
    });
```

Updated3 months ago

What’s Next

- iOS
- Android
- Sample Applications
- Table of Contents
- - SDK Download
- - Android SDK
- - - - Requirements
-