

expressPay SDK for Android Package

=====

This package contains the expressPay library, which allows you to integrate expressPay into your Android app and a sample app which was built on Android Studio

expressPay SDK for Android REQUIREMENTS

Android API >= 10

INSTALLATION

- 1. Open your project in Android Studio*
- 2. Download and unzip the library*
- 3. Add the expressPayLibrary-1.0.aar to your libs folder*
- 4. Open your app's build.gradle and add the "repositories" section:*

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

- 5. Also add this in the "dependencies" section:*

```
dependencies{
    //...
    compile (name: 'expressPayLibrary-1.4', ext:'aar')
    compile 'com.mcxiaoke.volley:library:1.0.19'

}
```

Get a token

Your server is responsible for generating a token, which contains the authorization and configuration details that your client needs to initialize the client SDK. We have include a sample server implemented in php

Initialization

```
/**
 *
```

```

    * Initialize a expressPayApi instance to communicate with
    expressPay SDK.
    *
    * @param context
    * @param yourServerURL the full path url to the location on
    your servers where you implement our server side sdk.
    * if null it defaults to https://
    sandbox.expresspaygh.com/api/sdk/php/server.php
    */
public ExpressPayApi(Context context,String yourServerURL)

```

```

public static interface ExpressPayPaymentCompletionListener {
    /**
     * @param paymentCompleted {@code true} if payment was successful,
     {@code false} otherwise
     * @param errorMessage the message if paymentCompleted is {@code
     false} or {@code null}
     * otherwise.
     */
    void onExpressPayPaymentFinished(boolean paymentCompleted, String
    errorMessage);
}

```

```

public static interface ExpressPaySubmitCompletionListener {

    /**
     * Called when a submit request is completed
     * @param response if the submit was successful it contains the
     results else or {@code null}
     * @param errorMessage error message for failure
     */

    void onExpressPaySubmitFinished(JSONObject response, String
    errorMessage);
}

```

```

public static interface ExpressPayQueryCompletionListener {

    /**
     * Called when a query request is completed
     * @param paymentSuccessful Boolean value of the transcation
     status
     * @param response result from query request or {@code null} if
     error
     * @param message message from query request or error message
     */
}

```

```

        void onExpressPayQueryFinished(Boolean paymentSuccessful,
JSONOobject response, String message);
}

```

```

/**
 * Create the whole checkout flow to display a checkout page for the
user to enter complete transaction
 * Will get a token and depend display the payment method views
 *
 * @param params Param required to start the checkout process.
 * @param activity The {@link Activity} used to perform the checkout.
 * @param listener to get results of the request
 */

```

```

public void submitAndCheckout(HashMap<String, String> params, final
Activity activity, final ExpressPayPaymentCompletionListener listener)

```

```

/**
 * Submit is called to initialise the you would like to request a
checkout on a users account.
 * The the default url is https://sandbox.expresspagh.com/api/sdk/php/
server.php
 *
 * @param params Hashmap<string, string> of information with the
following data
 * **@param string currency
 *         Currency of the transaction
 * ** @param string amount
 *         Amount the customer is paying for the order
 * ** @param string order_id
 *         Unique order identification number
 * **@param string order_desc
 *         Description of the order
 * **@param string account_number
 *         Customer account number at Merchant
 * **@param string redirect_url
 *         URL that customer should be redirected at the completion
of the payment process [ optional ]
 * **@param string order_img_url
 *         Image that customer should be shown at Checkout
[ optional ]
 * **@param string first_name
 *         Customer First name [ optional ]
 * **@param string last_name
 *         Customer Last name [ optional ]
 * **@param string phone_number
 *         Customer Phone Number [ optional ]

```

```

    * **@param string email
    * Customer email address
    *
    * @param activity The {@link Activity} used to perform the submit.
    Must implement ExpressPaySubmitCompletionListener
    * or an exception with be thrown
    * @param listener Listener would be called when the submit is
    completed

    */
    public void submit(HashMap<String, String> params, Activity activity,
    final ExpressPaySubmitCompletionListener listener)

    /**
    * Called to checkout the user
    * This method should only be called if you call the {@link package
    com.expresspaygh.api.ExpressPayApi.submit()}.
    * Once payment is complete the supplied
    * {@link package
    com.expresspaygh.api.ExpressPayApi.ExpressPayPaymentFinishedListener}
    will receive
    * a call to
    {@link com.expresspaygh.api.ExpressPayApi.ExpressPayPaymentCompletionLi
    stener#onExpressPayPaymentFinished(boolean, String)}
    *
    *
    *
    * @param activity The {@link Activity} used to perform the checkout.
    Must implement ExpressPayPaymentCompletionListener
    * or an exception with be thrown

    */
    public void checkout( Activity activity)

    /**
    * Called to begin payment. Once payment is complete the supplied
    * {@link package
    com.expresspaygh.api.expresspayapiExpressPayApi.ExpressPayPaymentFinis
    hedListener} will receive
    * a call to {@link
    com.expresspaygh.api.ExpressPayApi.ExpressPayPaymentCompletionListener
    #onExpressPayPaymentFinished(boolean, String)}
    * with an instance of a {@link com.expresspaygh.api.ExpressPayApi} or
    an error.
    *
    *
    * @param activity The {@link Activity} used to perform the checkout.
    Must implement ExpressPayPaymentCompletionListener
    * or an exception with be thrown

```

```

    * @param clientToken The client token obtained from a expressPay
server SDK.
    * @param redirectUrl A redirect url provided to expressPay server
side SDK.

    */
public void checkout( Activity activity, String clientToken, String
redirectUrl)

/**
 * Called to the query the status of transaction
 *
 *
 * @param token the activty used to perform the query has
 * @param listener Will be called when completion of query
 */
public void query( String token, final
ExpressPayQueryCompletionListener listener)

/**
 * Handles processing of any ExpressPayApi associated activity results
and posts
 *
 * @param activity The activity that received the {@link
Activity#onActivityResult(int, int, Intent))}
 * @param requestCode The requestCode received in {@link
Activity#onActivityResult(int, int, Intent))}
 * @param responseCode The responseCode received in {@link
Activity#onActivityResult(int, int, Intent))}
 * @param data The {@link Intent} received in {@link
Activity#onActivityResult(int, int, Intent))}
 */
public void onActivityResult(Activity activity, int requestCode, int
responseCode, Intent data)

/*
    Getter function to get the order id of the payment
    Will be set after a successful payment else will be null
 */
public String getOrderId()

/*
    Getter function to get the token of the payment
    Default will be the client token obtained from a expressPay server
side SDK
 */
public String getToken()

**
 * Set the development mode. by defaults it set to Debug. Please

```

change before moving to production

** @param value*

**/*

public void setDebugMode(**boolean** value)

GIVE FEEDBACK

Please report any bugs to integration@expresspaygh.com