

# Integration With PayPal's Mobile Payments Library

# Conceptual Overview

- **Goal:** Use PayPal for peer to peer payments within an Android application

# Conceptual Overview

- **Goal:** Use PayPal for peer to peer payments within an Android application
- We need to...
  - Decide what PayPal library to use

# Conceptual Overview

- **Goal:** Use PayPal for peer to peer payments within an Android application
- We need to...
  - Decide what PayPal library to use
  - Set up PayPal Developer and Sandbox accounts

# Conceptual Overview

- **Goal:** Use PayPal for peer to peer payments within an Android application
- We need to...
  - Decide what PayPal library to use
  - Set up PayPal Developer and Sandbox accounts
  - Integrate PayPal into an Android application

# MPL vs Mobile SDK

- PayPal offers two options for native Android Integration

# MPL vs Mobile SDK

- PayPal offers two options for native Android Integration
  - Mobile Payments Library (MPL)
  - Android Mobile SDK

# MPL vs Mobile SDK

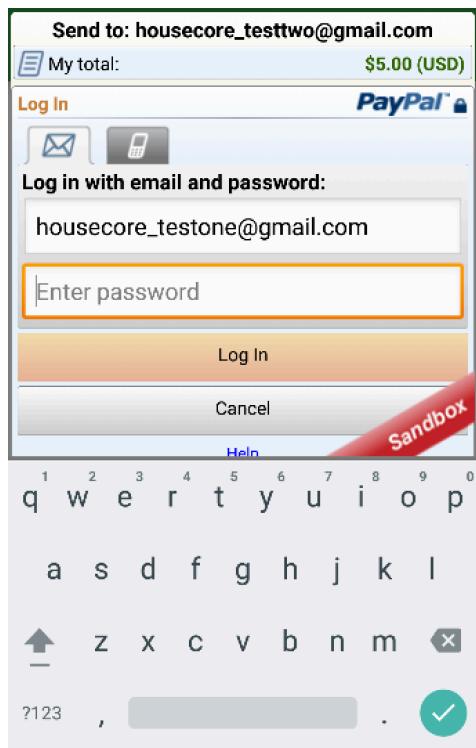
- PayPal offers two options for native Android Integration
  - Mobile Payments Library (MPL)
  - Android Mobile SDK
- Why pick one over the other?

# MPL vs Mobile SDK

- MPL
- Mobile SDK

# MPL vs Mobile SDK

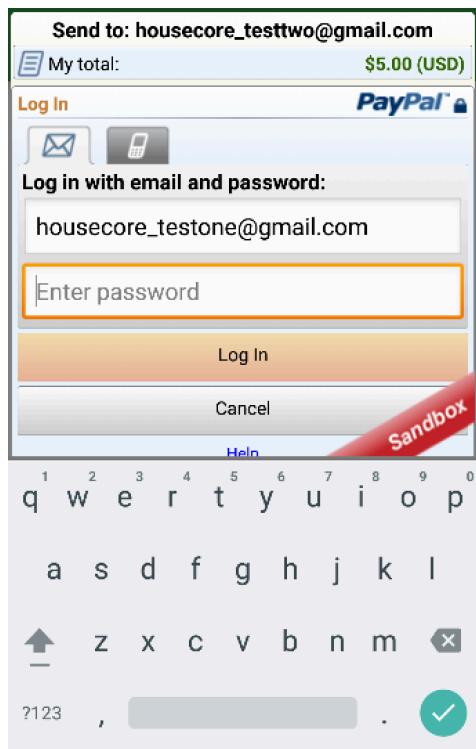
- MPL
  - Slightly outdated UI
- Mobile SDK



# MPL vs Mobile SDK

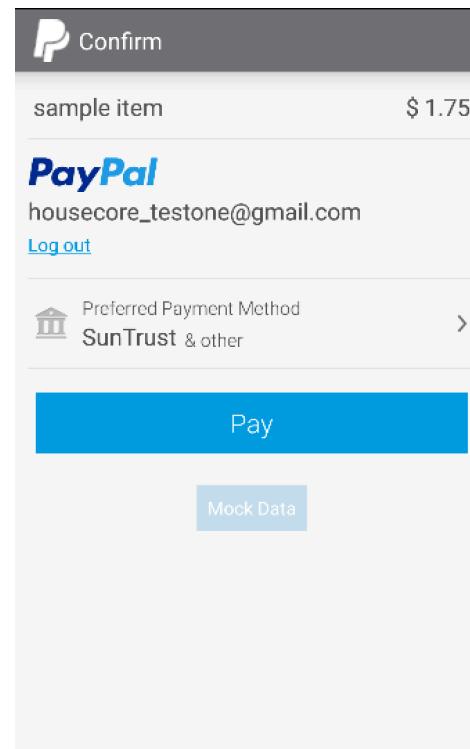
## ■ MPL

- Slightly outdated UI



## ■ Mobile SDK

- Modern sleek UI



# MPL vs Mobile SDK

- MPL
  - Slightly outdated UI
  - No new development
- Mobile SDK
  - Modern sleek UI

Mar 3, 2013 – Apr 19, 2016

Contributions to gh-pages, excluding merge commits



# MPL vs Mobile SDK

- MPL
  - Slightly outdated UI
  - No new development
- Mobile SDK
  - Modern sleek UI
  - Regularly Updated

Mar 3, 2013 – Apr 19, 2016  
Contributions to gh-pages, excluding merge commits

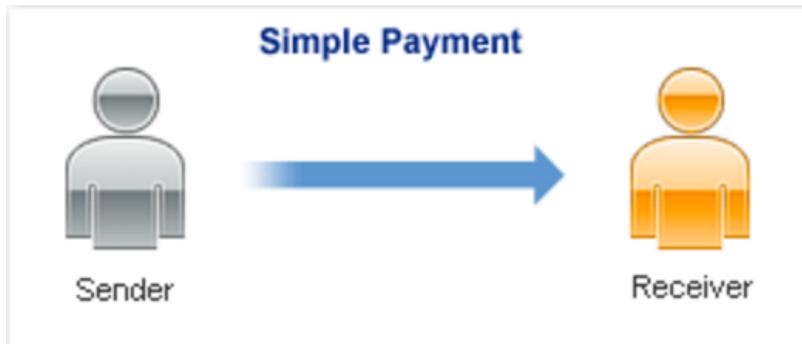


May 12, 2013 – Apr 19, 2016  
Contributions to master, excluding merge commits



# MPL vs Mobile SDK

- MPL
  - Slightly outdated UI
  - No new development
  - Support for P2P payment
- Mobile SDK
  - Modern sleek UI
  - Regularly Updated



# MPL vs Mobile SDK

- MPL
  - Slightly outdated UI
  - No new development
  - Support for P2P payment
- Mobile SDK
  - Modern sleek UI
  - Regularly Updated
  - No support for P2P payment



# MPL vs Mobile SDK

- MPL
  - Slightly outdated UI
  - No new development
  - Support for P2P payment
- Mobile SDK
  - Modern sleek UI
  - Regularly Updated
  - No support for P2P payment

Our use case requires Peer to Peer Payments so...



# MPL vs Mobile SDK

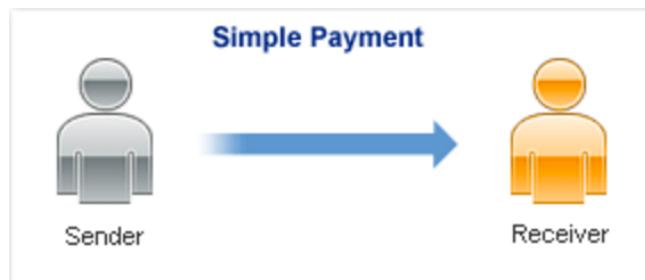
## ■ MPL

- Slightly outdated UI
- No new development
- Support for P2P payment

## ■ Mobile SDK

- Modern sleek UI
- Regularly Updated
- No support for P2P payment

Our use case requires Peer to Peer Payments so...



# Getting Started

- Create a PayPal developer account

<https://developer.paypal.com/>

# Getting Started

- Create a PayPal developer account

<https://developer.paypal.com/>

- Create a new REST API App

<https://developer.paypal.com/developer/applications/>

## REST API apps

Create an app to receive REST API credentials for testing and live transactions.

**Note** Features available for live transactions are listed in your [account eligibility](#).

[Create App](#)

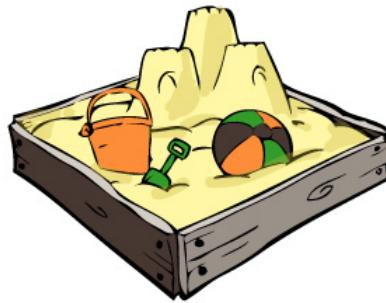
App name

HouseCore



# Getting Started

- Create a PayPal developer account  
<https://developer.paypal.com/>
- Create a new REST API App  
<https://developer.paypal.com/developer/applications/>
- Create a few Sandbox Test Accounts  
<https://developer.paypal.com/developer/accounts/>



# Getting Started

- Sandbox Test Accounts
  - Can be configured to test different scenarios

## Sandbox Test Accounts

[Create Account](#)

Questions? Check out the [Testing Guide](#). Non-US developers should read our [FAQ](#).

Want to link existing Sandbox Account with your developer account? [Click Here](#) and provide credentials of your sandbox account.

Total records: 4

<input type="checkbox"/>	Email Address	Type	Country	Date Created	
<input type="checkbox"/>	▶ housecore_testtwo@gmail.com	PERSONAL	US	31 Mar 2016	
<input type="checkbox"/>	▶ housecore_testone@gmail.com	PERSONAL	US	31 Mar 2016	
<input type="checkbox"/>	▶ jcollin6-facilitator@umd.edu	BUSINESS	US	31 Mar 2016	
<input type="checkbox"/>	▶ jcollin6-buyer@umd.edu	PERSONAL	US	31 Mar 2016	

# Getting Started

## ■ Sandbox Test Accounts

- Can be configured to test different scenarios
- Use fake money but real API calls!

### Sandbox Test Accounts

[Create Account](#)

Questions? Check out the [Testing Guide](#). Non-US developers should read our [FAQ](#).

Want to link existing Sandbox Account with your developer account? [Click Here](#) and provide credentials of your sandbox account.

Total records: 4

<input type="checkbox"/>	Email Address	Type	Country	Date Created	
<input type="checkbox"/>	▶ housecore_testtwo@gmail.com	PERSONAL	US	31 Mar 2016	
<input type="checkbox"/>	▶ housecore_testone@gmail.com	PERSONAL	US	31 Mar 2016	
<input type="checkbox"/>	▶ jcollin6-facilitator@umd.edu	BUSINESS	US	31 Mar 2016	
<input type="checkbox"/>	▶ jcollin6-buyer@umd.edu	PERSONAL	US	31 Mar 2016	

# Setting Up With Android Studio

- Download the PayPal MPL .jar file from  
[https://github.com/paypal/sdk-packages/  
tree/gh-pages/MPL](https://github.com/paypal/sdk-packages/tree/gh-pages/MPL)

# Setting Up With Android Studio

- Download the PayPal MPL .jar file from  
[https://github.com/paypal/sdk-packages/  
tree/gh-pages/MPL](https://github.com/paypal/sdk-packages/tree/gh-pages/MPL)
- Import the .jar file as a library



# Setting Up With Android Studio

- Download the PayPal MPL .jar file from  
[https://github.com/paypal/sdk-packages/  
tree/gh-pages/MPL](https://github.com/paypal/sdk-packages/tree/gh-pages/MPL)
- Import the .jar file as a library
- That's It! We're ready integrate the MPL into  
an application.

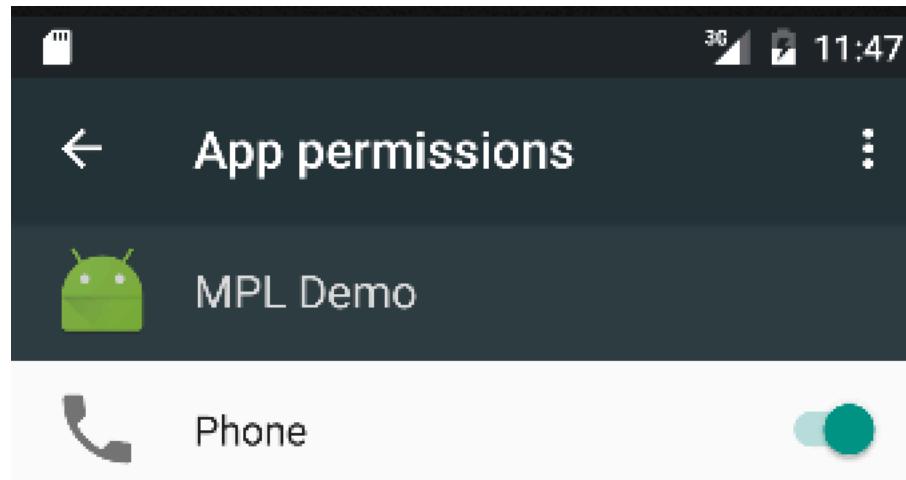
# Application Integration

- Add permissions to AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

# Application Integration

- Add permissions to AndroidManifest.xml
  - Make sure to grant application permission!



# Application Integration

- Add permissions to AndroidManifest.xml
  - Make sure to grant application permission!
- Add PayPalActivity to AndroidManifest.xml

```
<activity android:name="compaypal.android.MEP.PayPalActivity"  
    android:theme="@android:style/Theme.Translucent.NoTitleBar">  
</activity>
```

# Application Integration

- Add permissions to AndroidManifest.xml
  - Make sure to grant application permission!
- Add PayPalActivity to AndroidManifest.xml
- Import whatever PayPal methods you need in an Activity

```
import compaypalandroidMEPCheckoutButton;
import compaypalandroidMEPPayPal;
import compaypalandroidMEPPayPalPayment;
```

# Application Integration

- Add permissions to AndroidManifest.xml
  - Make sure to grant application permission!
- Add PayPalActivity to AndroidManifest.xml
- Import whatever PayPal methods you need in an Activity
- Declare application specific variables

```
// The PayPal environment to be used - can also be ENV_NONE and ENV_LIVE.  
private static final int environment = PayPal.ENV_SANDBOX;  
// The ID of your application that you received from PayPal. This is the default sandbox ID right now.  
private static final String appID = "APP-80W284485P519543T";  
// This is passed in for the startActivityForResult() android function, the value used is up to you.  
private static final int request = 1;
```

# Application Integration

- Add permissions to AndroidManifest.xml
  - Make sure to grant application permission!
- Add PayPalActivity to AndroidManifest.xml
- Import whatever PayPal methods you need in an Activity
- Declare application specific variables
- Initialize PayPal within the Activity

# Application Integration (Cont'd)

- Initialize PayPal within an Activity
  - Don't do this on the main thread

```
private void initLibrary() {  
    // Only need to initialize the library once.  
    PayPal pp = PayPal.getInstance();  
    if (pp == null) {  
        // Initialize PayPal  
        pp = PayPal.initWithAppID(this, appID, environment);  
        // Sender will pay fees. By default this is the receiver.  
        // It's free within the U.S. to send money to family and friends when you use only your PayPal  
        // balance or bank account, or a combination of their PayPal balance and bank account.  
        // So this really shouldn't come into play in our app.  
        pp.setFeesPayer(PayPal.FEEPAYER_SENDER);  
        // Shipping for Peer to Peer payments is pointless.  
        pp.setShippingEnabled(false);  
    }  
}
```

# Application Integration (Cont'd)

- Initialize PayPal within an Activity
  - Don't do this on the main thread

Initialization will fail if App permissions are not granted!

```
private void initLibrary() {  
    // Only need to initialize the library once.  
    PayPal pp = PayPal.getInstance();  
    if (pp == null) {  
        // Initialize PayPal  
        pp = PayPal.initWithAppID(this, appID, environment);  
        // Sender will pay fees. By default this is the receiver.  
        // It's free within the U.S. to send money to family and friends when you use only your PayPal  
        // balance or bank account, or a combination of their PayPal balance and bank account.  
        // So this really shouldn't come into play in our app.  
        pp.setFeesPayer(PayPal.FEEPAYER_SENDER);  
        // Shipping for Peer to Peer payments is pointless.  
        pp.setShippingEnabled(false);  
    }  
}
```



# Application Integration (Cont'd)

- Initialize PayPal within an Activity
  - Don't do this on the main thread
- Add a PayPal button

```
PayPal pp = PayPal.getInstance();
// Get the CheckoutButton. There are five different sizes.
// The text on the button can either be of type TEXT_PAY or TEXT_DONATE.
launchSimplePayment = pp.getCheckoutButton(this, PayPal.BUTTON_194x37, CheckoutButton.TEXT_PAY);
// You'll need to have an OnClickListener for the CheckoutButton.
// For this application, PayActivity implements OnClickListener and we
// have the onClick() method below.
launchSimplePayment.setOnClickListener(this);
// The CheckoutButton is an android LinearLayout so we can add it to our display like any other View.
```



Pay with **PayPal**™

# Application Integration (Cont'd)

- Initialize PayPal within an Activity
  - Don't do this on the main thread
- Add a PayPal button
- Create a payment object in click handler

```
// Create a basic PayPalPayment.  
PayPalPayment payment = new PayPalPayment();  
payment.setSubtotal(PAYMENT_AMOUNT);  
// Sets the currency type for this payment.  
payment.setCurrencyType("USD");  
// Sets the recipient for the payment. This can also be a phone number.  
payment.setRecipient(RECIPIENT_EMAIL);  
// Depending on the payment type the checkout experience can vary slightly.  
// Set this to personal to avoid fee payments.  
payment.setPaymentType(PayPal.PAYMENT_TYPE_PERSONAL);
```

# Application Integration (Cont'd)

- Initialize PayPal within an Activity
  - Don't do this on the main thread
- Add a PayPal button
- Create a payment object in click handler

We defined these using Intent extras but you can define them however you like



```
// Create a basic PayPalPayment.
PayPalPayment payment = new PayPalPayment();
payment.setSubtotal(PAYMENT_AMOUNT);
// Sets the currency type for this payment.
payment.setCurrencyType("USD");
// Sets the recipient for the payment. This can also be a phone number.
payment.setRecipient(RECIPIENT_EMAIL);
// Depending on the payment type the checkout experience can vary slightly.
// Set this to personal to avoid fee payments.
payment.setPaymentType(PayPal.PAYMENT_TYPE_PERSONAL);
```

# Application Integration (Cont'd)

- Initialize PayPal within an Activity
  - Don't do this on the main thread
- Add a PayPal button
- Create a payment object in click handler
- Checkout! From here the checkout activity takes care of all PayPal interaction

```
// Use checkout to create our Intent.  
Intent checkoutIntent = PayPal.getInstance().checkout(payment, this, new ResultDelegate(PAYMENT_ID));  
// Use the android's startActivityForResult() and pass in our Intent. This will start the library.  
startActivityForResult(checkoutIntent, request);
```

# Application Integration (Cont'd)

- Initialize PayPal within an Activity
  - Don't do this on the main thread
- Add a PayPal button
- Create a payment object in click handler
- Checkout! From here the checkout activity takes care of all PayPal interaction

Wait... What the heck is a ResultDelegate?

```
// Use checkout to create our Intent.  
Intent checkoutIntent = PayPal.getInstance().checkout(payment, this, new ResultDelegate(PAYMENT_ID));  
// Use the android's startActivityForResult() and pass in our Intent. This will start the library.  
startActivityForResult(checkoutIntent, request);
```



# Application Integration (Cont'd)

- The PayPalResultDelegate interface provides a way to be notified immediately when a payment is completed

# Application Integration (Cont'd)

- The PayPalResultDelegate interface provides a way to be notified immediately when a payment is completed
- Three possible completion states
  - Succeeded
  - Failed
  - Cancelled

# Demo Time!



<https://github.com/jbccollins/CMSC436-PayPal-MPL-Demo>