

INNOVATION IN MOBILE PAYMENTS



GDG DEVFEST

November 2012, Vienna

Tim Messerschmidt
Developer Evangelist
[@SeraAndroid](#)

Who am I?

- Passionate developer
- Mobile- & Web-Developer
- Android- & Twitter-Fanboy



Agenda

- What is PayPal Access?
 - Underlying techniques
- How to accept payments?
- Outlook: Innovation
 - Real world payments

Slides



goo.gl/u3Rix

SlideShare:
PayPalEUDevs

WHAT IS ACCESS?

Introduced in 2011

Can be used to login
with your existing
PayPal credentials



Figure:
Q3 2012
active
users

Leverage existing
technology to push
your own service(s)

Based on OAuth 2.0 & OpenID Connect

**Not related to
payments**

Free to use

Registration of apps: devportal.x.com

Privacy Policy URL:

(Recommended)

Protocols [?]:*

☐ OpenID 2.0

☒ OAuth 2.0 / Open Id Connect

OAuth 2.0 / OpenID Connect Attributes

Return URL:*

Attributes Level:*

☐ Full Name

☐ Gender

☐ Date of Birth

☐ Time Zone

☐ Locale

☐ Language

☐ Email Address

☐ Street Address

☐ State

☐ Zip

☐ Country

☐ Phone Number

☐ Age Range

☐ Account Type

☐ Account Verified

☐ Account Creation Date

**Additional features
coming soon!**

HOW DOES IT LOOK LIKE?

HAVE a little pet?

[click here](#) TO READ
ABOUT FEED



CAT's food NEW PRODUCTS

[VIEW NOW!](#)



[HOME PAGE](#)
[NEW PRODUCTS](#)
[SPECIALS](#)
[MY ACCOUNT](#)
[CONTACT US](#)
CATEGORIES:

- [Advantage Flea](#)
- [Advantix K9](#)
- [Dog House](#)
- [Airline Carriers](#)
- [Allergies: Human](#)
- [Andis Clippers](#)
- [Anti-bark Devices](#)
- [Jingle Collars](#)
- [Backpacks](#)
- [Bad Breath](#)
- [Balls](#)
- [Bandannas](#)
- [Bark Collars](#)
- [Bath Supplies](#)
- [Boomboxes](#)

FEATURED PRODUCTS:
Cat Print Tapestry Kitty Cave

\$39.99
[details](#)
[add to cart](#)
Lipiderm Healthy Skin and Coat Gel Capsules

\$17.99
[details](#)
[add to cart](#)
MANUFACTURERS:

INFORMATION:

- [No more scratching](#)
- [Your dog and you](#)
- [Happy pet owners](#)

SPECIALS:
KMR Powder by Pet Ag

\$10.99
\$10.99

save **20%**
MORE THAN 1000 items

[CLICK
HERE](#)


Log in with PayPal Access

We will not share your financial information with anyone. We use your PayPal account to log you in with tinypay.me .

Email address

PayPal password


[Forgot your email address or password?](#)

Don't have an account? [create one now](#)


Log In

[Cancel](#)

PayPal protects your privacy and security.

Allow Access**PayPal.com** → **tinypay.me****To complete login with tinypay.me , agree to:** PayPal sharing the following information:

Full Name	Tim Messerschmidt
Email address	timmesserschmidt@googlemail.com
Verified	Yes

 The user agreement and privacy policy of tinypay.me**Allow**[Cancel](#)

See also: [User Agreement](#) | [Privacy Policy](#) | [Key Payment and Service Information](#)

HOW DOES IT WORK?



OAuth?

OpenID?

OpenID Connect?

OAuth 1.0

OAuth 2.0

OPINIONS ON OAUTH 2

OAuth 2.0 & the Road to Hell

*“OAuth 2.0 offers
little to none code
reusability”*

*“What 2.0 offers is a
blueprint for an
authorization protocol”*

On the Deadness of OAuth 2

*“OAuth 2 is
useful today.”*

“OAuth 2 may not be perfect, and may have been harmed by the Enterprise crap, but the core of Web functionality [...] seems to have survived.”

OpenID Connect

5 scopes
for access
to the
profile:

1. profile
2. email
3. address
4. phone
5. attributes

THE DIFFERENCE

OAuth 2.0 implementation can be easily changed to OpenID Connect

Jonathan LeBlanc: <https://www.x.com/developers/community/blogs/jcleblanc/migrating-paypal-access-integration-oauth-2-openid-connect>

**WHY
SHOULD I
USE THIS?**

People forget passwords...

“45 % admit to leaving a website instead of re-setting their password or answering security questions” *

* Blue Inc. 2011

People don't like to register...

Out of **657** surveyed users **66 %** think that social sign-in is a desirable alternative. *

* Blue Inc. 2011

THE VALUE

Verified user accounts

Leverage an existing profile

```
{
  "status" : "SUCCESS",
  "statusMessage" : "ACTIVE",
  "identity" : {
    "language" : "en_US",
    "firstName" : "John",
    "lastName" : "Smith",
    "fullName" : "John D Smith",
    "emails" : [ "jsmith@x.com" ],
    "addresses" : [ {
      "state" : "CA",
      "country" : "US",
      "street1" : "1234 street",
      "street2" : "Suite #250",
      "city" : "San Jose",
      "zip" : "12345"
    } ],
    "telephoneNumber" : "123-456-7890",
    "dob" : "1980-01-01",
    "timezone" : "PDT",
    "gender" : "M"
  }
}
```


THE FLOW AND THE CODE

Authorization Flow

Client

1. Open Authorization Endpoint URL
4. Check callbacks for Authorization Token
5. Request a valid Access Token
7. Retrieve user's resources

Server

2. Provide login page
3. Return Authorization Token after successful login
6. Check Authorization Token & return Access Token if valid

Your components (OAuth 2)

Server endpoints:

```
URL_AUTHORIZE      = "https://identity.x.com/xidentity/resources/authorize";
URL_TOKENSERVICE = "https://identity.x.com/xidentity/oauthtokenservice";
URL_PROFILE        = "https://identity.x.com/xidentity/resources/profile/me";
```

Client details:

```
/*
 * The id and secret are used to identify your application.
 * Those credentials can be obtained at https://devportal.x.com/
 */

CLIENT_ID      = "EFS0iRFhehCj_hhJATYIs01Xx5vEouCoK7MkVInoxsF8";
CLIENT_SECRET  = "EN0Yaf7NJNF1bbd8kghckKq5wrbwxHeTcY7_U8Ay43nWWk-DXgWYgT8-sVVKhns0PQ";
```

Load the Authorization URL
in a WebView and...

... start checking the URLs your
WebView is loading

```
@Override
public boolean shouldOverrideUrlLoading(WebView view, String url) {
    if (url.startsWith("http://myReturnUrl.com/auth")) {
        getAccessToken(url);
        return true;
    }
    return super.shouldOverrideUrlLoading(view, url);
}
```

Retrieve the Access Token

```
private void getAccessToken(String url) {
    final Uri uri = Uri.parse(url);
    final String code = uri.getQueryParameter("code");
    final String urlParams = helper.getTokenServiceParameters(code);
    final String urlString = helper.getTokenServiceUrl();

    new AsyncConnection(new AsyncConnectionListener() {
        public void connectionDone(String result) {
            try {
                final JSONObject object = new JSONObject(result);
                final String accessToken = object
                    .getString("access_token");

                if (accessToken != null && !accessToken.equals("")) {
                    getProfile(accessToken);
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }).execute(AsyncConnection.METHOD_POST, urlString, urlParams);
}
```

THE REPLY

```
{  
  "access_token": "something not so long",  
  "token_type": "Bearer",  
  "refresh_token": "something not so long",  
  "expires_in": 900,  
  "id_token": "something very long"  
}
```

Conclusion

PayPal Access enhances applications by adding a verified user-base

PAYMENTS

2 ways: **Native or Web**

MPL

=

**Mobile Payments
Library**

Add the .jar to your
project's libs folder

Add these permissions to your project:

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

Add this activity to your project:

```
<activity  
    android:name="com.paypal.android.MEP.PayPalActivity"  
    android:configChanges="keyboardHidden|orientation"  
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
```

Initialize the library in a separate Thread

```
PayPal pp = PayPal.getInstance();  
if (pp == null) {  
    pp = PayPal.initWithAppID(MainActivity.this,  
        appID, PayPal.ENV_SANDBOX);  
}
```

Add the PayPal button to your existing layout

```
View launchSimplePayment = pp.getCheckoutButton(MainActivity.this,  
    PayPal.BUTTON_294x45, CheckoutButton.TEXT_PAY);  
launchSimplePayment.setOnClickListener(MainActivity.this);  
contentView.addView(launchSimplePayment);
```


Create the payment

```
PayPalPayment payment = new PayPalPayment();  
payment.setCurrencyType("EUR");  
payment.setRecipient("tim@fancyunicorn.com");  
payment.setPaymentType(PayPal.PAYMENT_TYPE_GOODS);  
payment.setSubtotal(new BigDecimal("8.25"));  
Intent checkoutIntent =  
    PayPal.getInstance().checkout(payment, this, new Delegate());  
startActivityForResult(checkoutIntent, PAY);
```

React to the result

```
if(requestCode == PAY) {  
    if(resultCode == RESULT_OK) {  
        Toast.makeText(this, "Success!",  
            Toast.LENGTH_LONG).show();  
    } else if(resultCode == RESULT_CANCELED) {  
        Toast.makeText(this, "Payment cancelled!",  
            Toast.LENGTH_LONG).show();  
    }  
}
```

Done!



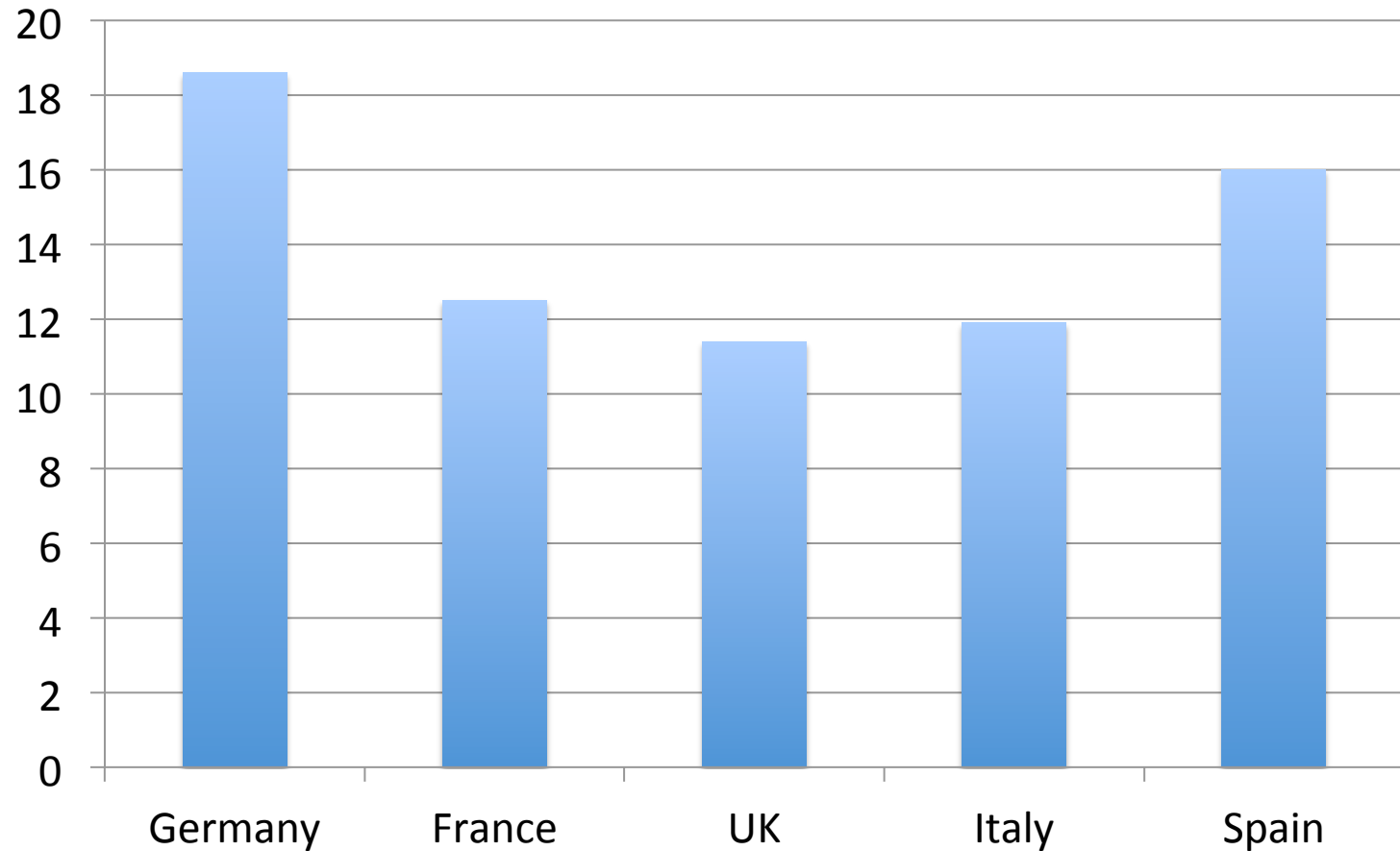
Conclusion

*Payments must not be
complicated*

INNOVATION

QR Code adoption between different countries

% of Smartphone Audience



* comScore MobiLens July 2012

Adoption of QR Codes

70% of 30.000.000 surveyed
German households recognize QR
Codes and know how to use them *

* Nielsen 2011

PayPal / QRShopping



Jetzt direkt kaufen!

Available
for both
Android
& iOS

FURTHER INFORMATION

Useful links

- **goo.gl/y9HKO**
 - Migrating PayPal Access from OAuth 2 to OpenID Connect (Jonathan LeBlanc)
- **goo.gl/1wjRV**
 - Sample project which has some helper classes that enable easy integration Access into your Android app
 - Apache V2

Official developer resources

- **x.com/identity**
 - PayPal Access Developer Guide
- **x.com/mobile**
 - PayPal payment products
 - Mobile Payments Library (native)
 - Mobile Express Checkout (web)

Help!!?! Problems?

- **paypal.com/dts**
 - Developer Technical Services
 - Ticketing
- **x.com/developers/paypal/forums**
 - PayPal Developer Forums

QUESTIONS?

THANKS!

tmesserschmidt@paypal.com
@seraandroid / @paypaleurodev
+tim messerschmidt
slideshare.net/PayPalEUDevs