
Guia de Integração

Biblioteca Android PagSeguro UOL - Checkout in App Manual de Uso

Histórico de Versões

- 0.0.1 : **Versão inicial** - 13/07/2016

Copyright

Todos os direitos reservados. O UOL é uma marca comercial do UNIVERSO ONLINE S / A. O logotipo do UOL é uma marca comercial do UNIVERSO ONLINE S / A. Outras marcas, nomes, logotipos e marcas são de propriedade de seus respectivos proprietários. As informações contidas neste documento pertencem ao UNIVERSO ONLINE S/A. Todos os direitos reservados. UNIVERSO ONLINE S/A. - Av. Faria Lima, 1384, 6º andar, São Paulo / SP, CEP 01452-002, Brasil. O serviço PagSeguro não é, nem pretende ser comparável a serviços financeiros oferecidos por instituições financeiras ou administradoras de cartões de crédito, consistindo apenas de uma forma de facilitar e monitorar a execução das transações de comércio eletrônico através da gestão de pagamentos. Qualquer transação efetuada através do PagSeguro está sujeita e deve estar em conformidade com as leis da República Federativa do Brasil. Aconselhamos que você leia os termos e condições cuidadosamente.

Aviso Legal

O UOL não oferece garantias de qualquer tipo (expressas, implícitas ou estatutárias) com relação às informações nele contidas. O UOL não assume nenhuma responsabilidade por perdas e danos (diretos ou indiretos), causados por erros ou omissões, ou resultantes da utilização deste documento ou a informação contida neste documento ou resultantes da aplicação ou uso do produto ou serviço aqui descrito. O UOL reserva o direito de fazer qualquer tipo de alterações a quaisquer informações aqui contidas sem aviso prévio.

Visão Geral

A biblioteca Checkout in App tem como foco auxiliar desenvolvedores que desejam prover em seus aplicativos toda a praticidade e segurança fornecida pelo PagSeguro no segmento de pagamentos móveis através de smartphones e tablets. Para ajudar a entender como a biblioteca pode ser utilizada, apresentamos o seguinte cenário:

• Cenário Exemplo: Solução de pagamentos com Checkout in App. A empresa X desenvolve um aplicativo para seus clientes permitindo-os efetuar pagamento de serviços prestados ou itens (produtos) vendidos. Neste cenário o aplicativo da empresa X faz uso da biblioteca PagSeguro "Checkout in App" autorizando a Library com a sua conta PagSeguro (E-mail vendedor e Token referente da conta). Os clientes da empresa X que utilizam o aplicativo para realizar o pagamento em um ambiente seguro para autenticação do usuário utilizando uma conta PagSeguro (usuário comprador). Após autenticação o usuário do aplicativo da empresa X poderá realizar pagamentos utilizando sua conta PagSeguro (usuário comprador). A empresa X receberá os pagamentos em sua conta PagSeguro configurada como vendedor na Lib Checkout in App.

Conceitos Básicos

Antes de fazer uso da biblioteca é importante que o desenvolvedor realize alguns procedimentos básicos, além de assimilar alguns conceitos importantes para o correto funcionamento de sua aplicação. É necessário ter em mãos o token da conta PagSeguro que será configurado como vendedor (Seller), tal token pode ser obtido no ibanking do PagSeguro. (Vide tópico abaixo).

Obtendo Token da conta PagSeguro

Para realizar transações utilizando a biblioteca é necessária uma conta PagSeguro. Caso não tenha uma Acesse: www.pagseguro.com.br. Com a conta PagSeguro criada é necessário ter o Token da conta que será utilizada na configuração como vendedor na Library Checkout in App.

OBTENDO TOKEN DA CONTA PAGSEGURO PARA INTEGRAÇÃO COM API's Na pagina do ibanking do PagSeguro em sua conta:

- 1- Click na guia "**Minha Conta**";
 - 2- No Menu lateral clique em "**Preferências**";
 - 3- Pressione o botão "**Gerar Token**";
 - 4- Armazene esse **TOKEN** em algum lugar pois iremos utilizá-lo a seguir nesse guia de integração.
-

Requisitos Mínimos

Antes de iniciar o guia de integração com a Library Checkout in App é válido ressaltar alguns requisitos mínimos para funcionamento da Biblioteca.

Versão Android: Android 4.x+;

Aplicativo deve fazer uso da biblioteca de compatibilidade da Google Support Library
AppCompact * v7;

Conexão com internet;

Permissões no AndroidManifest.xml (Vide tópico abaixo);

Importar corretamente as dependências no .gradle;

Permissões AndroidManifest.xml

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

Instalação

Via gradle:

```
repositories {

    mavenCentral()

    maven {
        url "https://jitpack.io"
    }

    maven {
        credentials {
            username '<BITBUCKET EMAIL>'
            password '<BITBUCKET PASSWORD>'
        }
        url "https://bitbucket.org/uolinc/checkout-in-app-
android/raw/mvn-repo/repository/"
    }
}

dependencies {
    ...
    compile 'br.com.uol.pslibs:checkout-in-app:0.0.1'
    ...
}
```

Metodologia

Vejam os agora como integrar a biblioteca Checkout in App do PagSeguro em seu aplicativo Android.

Para utilizar a library **Checkout in App** do PagSeguro UOL para Android são necessários 6 etapas:

1 – Implementação base: Inicialização da Library e controles definidos durante o ciclo de vida da Activity;

2 – Autorização: **PSCheckout.init()** - Configurações das credenciais da conta PagSeguro como vendedor na Checkout in App;

3 – Pagamento: **PSCheckout.pay()** - Para utilizar o método de pagamento deve seguir uma estrutura de parametrização;

4 – Listagem de cartões: **PSCheckout.showListCards()** - A Lib Checkout in App faz gerenciamento dos seus cartões para serem utilizados em sua aplicação específica, gerenciando todo possível de análise de risco e fraudes de cartões, veja no tópico referente mais informações da utilização desse método;

5 – Cartão principal: **PSCheckout.getMainCard()** A Lib mantém um cartão definido como cartão principal. Esse cartão será utilizado no processo de pagamento acionado pelo método **PSCheckout.pay()**

6 – Logout: **PSCheckout.logout()** - Faz logout e apaga dados da sessão do usuário PagSeguro comprador logado na Lib Checkout in App;

1 - Implementação base

Antes de realizar as chamadas para realizar pagamento, temos de inicializar a biblioteca. Este processo deve ser realizado uma única vez durante o ciclo de vida da aplicação. Além disso, existem alguns métodos que devem ser chamados e eventos específicos de uma Activity, garantindo que a biblioteca possa controlar e liberar todos os recursos utilizados assim que necessário. Assumindo que sua aplicação utiliza uma Activity com diversos Fragments no processo de venda, você deve efetuar este procedimento conforme o código abaixo:

```
public class ExampleActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.example);  
    }  
}
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    //Fornece controle para LIB de Activity results
    PSCheckout.onActivityResult(this, requestCode, resultCode,
data); //Controle Lib Activity Life Cycle
}

@Override
public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
    //Android 6+ fornece controle para LIB para request de permissões
    PSCheckout.onRequestPermissionsResult(this, requestCode,
permissions, grantResults); //Controle Lib Activity Life Cycle
}

@Override
public void onBackPressed() {
    if (PSCheckout.onBackPressed(this)) { //Controle Lib Button back
        super.onBackPressed();
    }
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            PSCheckout.onHomeButtonPressed(this); //Controle Lib Home
Button
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    PSCheckout.onDestroy(); //Controle Lib Activity Life Cycle
}
}
```

Ao utilizar o código acima você habilita o uso da biblioteca enquanto a Activity em que o código reside esteja ativa bastando apenas efetuar as chamadas para as rotinas desejadas, é de extrema importância para o funcionamento da “Library” as chamadas dos métodos do ciclo de vida da Activity apontado no código mostrado acima.

2 - Autorização

Antes de utilizar os demais métodos da biblioteca é de extrema importância passar as configuração para inicialização da Lib Checkout in App.

Configuração de inicialização da Lib:

->(String)Seller Email (E-mail da conta que será utilizado como vendedor);

->(String)Seller Token (Token da conta que será utilizado como vendedor, foi explicado anteriormente nessa documentação como obter esse token);

-> (Int) Fragment container);

Exemplo de inicialização da Lib:

```
//Inicialização a lib com parametros necessarios
    PSCheckoutConfig psCheckoutConfig = new PSCheckoutConfig();
    psCheckoutConfig.setSellerEmail(SELLER_EMAIL);
    psCheckoutConfig.setSellerToken(SELLER_TOKEN);
    //Informe o fragment container
    psCheckoutConfig.setContainer(R.id.fragment_container);

    PSCheckout.init(getActivity(), psCheckoutConfig);
```

3 - Pagamento

Para realização do pagamento devemos utilizar o método **PSCheckout.pay()**, deve ser passado dois parâmetros nesse método:

PSCheckoutRequest -> Objeto que vai conter informações necessárias para o processamento do pagamento;

PSCheckout.PSCheckoutListener -> Esse listener será o ponto focal para validar o status das operações de pagamentos da Library Checkout in App.

Abaixo vamos demonstrar como funciona o objeto PSCheckoutRequest:

```
//Valor do produto / serviço
    BigDecimal amount = new BigDecimal(1.0);
    //quantidade de parcelas
    int quantityParcel = 1;
    //id do produto
    String productId = "001";
    //Descrição do produto
    String description = "Produto Exemplo";

    PSCheckoutRequest psCheckoutRequest =
        new PSCheckoutRequest().withReferenceCode("123")
            .withNewItem(description,
String.valueOf(quantityParcel), amount, productId);
```

Agora uma demonstração de utilização do PSCheckout.PSCheckoutListener:

```
/**
    * PSCheckout listener
```

```
* Este listener será o ponto focal para validar os resultados de sua
transação;
*/
PSCheckout.PSCheckoutListener psCheckoutListener = new
PSCheckout.PSCheckoutListener() {
    @Override
    public void onSuccess(PagSeguroResponse pagSeguroResponse, Context
context) {
        //Sucesso na transação
    }

    @Override
    public void onFailure(PagSeguroResponse pagSeguroResponse, Context
context) {
        //Falha na transação
        Log.v(TAG, "Falha na transação, " + " Erro: " +
pagSeguroResponse.getErrorCode() + ", " + pagSeguroResponse.getMessage());
    }

    @Override
    public void onProcessing(Context context) {
        //Listener para UI Thread
        //Ex. Dialogs, Loading, animações, etc;
    }
}
```

Exemplo chamada do método Pay():

```
PSCheckout.pay(psCheckoutRequest, psCheckoutListener);
```

4 - Listagem de cartões

A biblioteca Checkout in App do PagSeguro UOL fornece um gerenciamento de cartões de crédito tendo toda uma criticidade com análise de riscos e fraudes, gerenciando cartões da sua conta PagSeguro por aplicação.

Para acessar a parte de gerenciamento de cartões basta utilizar o método **PSCheckout.showListCards()**

Exemplo utilização do método:

```
PSCheckout.showListCards();
```

5 - Cartão Principal

A biblioteca trabalha mantendo sempre um cartão de crédito definido como principal, esse cartão será utilizado no processo de pagamento acionado pelo método **PSCheckout.pay()**, para recuperar a informação desse cartão principal utilizamos o método **PSCheckout.getMainCard()**.

Esse método retorna um objeto PSWalletMainCardVO como resposta, contendo os seguintes métodos:

```
//int getImageResource() -> Retorna uma imagem da bandeira do cartão principal;
//String getFinalCard() -> Retorna o final do cartão utilizado

//Exemplo utilização do metodo PSCheckout.getMainCard()

/* Obtem o cartão principal setado na lib, caso queira deixar visivel no App.
 *
 * @Return int ImgResource
 * @Return String Final cartão
 */
if (PSCheckout.getMainCard() != null) {
    imgFlagCard.setImageResource(PSCheckout.getMainCard().getImageResource());
    tvFinalCard.setText(PSCheckout.getMainCard().getFinalCard());
} else {
    Snackbar.make(getView(), "Sem cartão principal no momento...",
        Snackbar.LENGTH_SHORT).show();
}
```

6 - Logout

A Biblioteca Checkout in App fornece um método para logout ao qual limpa a sessão da conta PagSeguro logada da lib.

Exemplo de utilização do método PSCheckout.logout()

```
//Faz Logout do usuario na libray Checkout in App
PSCheckout.logout(getActivity());
//"Your APP" does things
//Aqui por exemplo como fizemos o logout estaremos limpando o cartão principal
imgFlagCard.setImageResource(R.drawable.ic_credit_card_grey600_48dp);
tvFinalCard.setText("Cartão");
Snackbar.make(getView(), "Logout", Snackbar.LENGTH_SHORT).show();
```

Obs: Os trechos utilizados nesta documentação foram retirados da aplicação de exemplo disponível nesse repositório.

Código de Erros

Abaixo seguem os códigos de erro que podem ser retornados pela biblioteca:

1009 - SESSION_EXPIRED (Sessão expirada, refaça a operação e realize login novamente.);

9000 - NETWORK_ERROR (Falha de conexão);

UOL - O melhor conteúdo

© 1996-2016 Todos os direitos reservados.

9001 - REFUSED_TRANSACTION_ERROR (Transação cancelada ou recusada);

9002 - CREATE_TRANSACTION_ERROR (Falha ao criar transação);

9003 - TIME_OUT_CHECK_TRANSACTION (Timeout verificação status da transação);

9004 - CHECK_TRANSACTION_ERROR (Falha na verificação da transação);

9005 - TIME_OUT_CHECK_TRANSACTION_VALIDATOR (Timeout verificação status da transação validadora);

Obs: Para acesso a um ENUM contendo esses erros mapeados utilize a classe:
ErrorCode.java

Customização dos componentes da Lib. Checkout in App

Com intuito de possibilitar uma ambientalização com o App do parceiro a nossa biblioteca traz um recurso muito prático para customizar nossos componentes. A maioria da componentização da nossa biblioteca foi feita seguindo a Guideline do Material Design da Google Android. Abaixo você vai aprender como é possível customizar a nossa biblioteca.

@color/ps_lib_checkbox_color -> Cor do background do checkbox

@color/ps_lib_checkbox_text_color -> Cor do texto do checkbox

@color/ps_lib_color_primary -> Cor da tool bar

@color/ps_lib_color_primary_dark -> Cor da status bar

@color/ps_lib_field_error -> Cor de erro do Fancy Field

@color/ps_lib_field_focus_off -> Cor do Fancy Field sem foco

@color/ps_lib_field_focus_on -> Cor do Fancy Field com foco

@color/ps_lib_primary_button_color -> Cor do background do botão

@color/ps_lib_primary_button_disable_text_color -> Cor do texto do botão desabilitado

@color/ps_lib_primary_button_text_color -> Cor do texto do botão

@color/ps_lib_primary_disable_button_color -> Cor do background do botão desabilitado

@color/ps_lib_text_link_button -> Cor do texto do botão estilo link

<!--Exemplo de customização:-->

<!-- Tema exemplo 1: Tons de marron -->

<!--Material status e tool bar-->

<color name="ps_lib_color_primary_dark">#8B4513</color>

```
<color name="ps_lib_color_primary">#A0522D</color>

<!-- FancyEditField colors-->
<color name="ps_lib_field_focus_off">#CCCCCC</color>
<color name="ps_lib_field_focus_on">#D2691E</color>
<color name="ps_lib_field_error">#D62C2E</color>

<!-- Checkbox colors -->
<color name="ps_lib_checkbox_color">#D2691E</color>
<color name="ps_lib_checkbox_text_color">#717171</color>

<!-- Button colors -->
<color name="ps_lib_primary_button_color">#D2691E</color>
<color name="ps_lib_primary_disable_button_color">#70D2691E</color>
<color
name="ps_lib_primary_button_disable_text_color">@color/ps_lib_color4</color
>
<color name="ps_lib_primary_button_text_color">#FFD700</color>

<!-- Link button -->
<color name="ps_lib_text_link_button">#D2691E</color>
```