

API DE PAGAMENTO POS DIGITAL GETNET

CLASSIFICAÇÃO DO DOCUMENTO – CONFIDENCIAL

Este documento pode conter informação confidencial e/ou privilegiada. Se você não for o destinatário ou a pessoa autorizada a receber este documento, não deverá utilizar, copiar, alterar, divulgar a informação nele contida ou tomar qualquer ação baseada nessas informações. Se você recebeu este documento por engano, por favor, avise imediatamente o responsável, devolvendo o documento em seguida. Agradecemos sua cooperação.

Sumário

Histórico de alterações	3
Introdução	3
Kit de Desenvolvimento	3
Hardwares disponíveis	4
Ingenico – APOS A8	4
Gertec - GPOS700	6
Impressão	7
O que você precisa saber antes de começar a desenvolver	8
Boas práticas	8
Requisitos mandatórios	8
Submetendo um Aplicativo	9
API	10
Simulador da API	10
Integração com a API de pagamento	10
Deeplinks disponíveis	11
Resultados	11
Funcionalidades	11
Pagamento	11
Pré-autorização	17
Estorno	19
Reimpressão	20
Info 21	
Dúvidas Frequentes	22

Histórico de alterações

Versão	Data	Autor	Descrição
1.0	26/04/2018	Vinicius P. Bohrer	Versão inicial
2.0	30/05/2018	Felipe Espitalher dos Passos	Mais campos no payment
2.1	21/06/2018	Vinicius P. Bohrer	Mais campos no refund e payment Mais um requisito mandatório
2.2	09/07/2018	Luiz Gustavo	Ajuste no campo type em payment
2.3	10/09/2018	Luiz Gustavo	Adicionado o terminal da Ingenico no cap. Hardware disponível

Introdução

O objetivo deste documento é apresentar tecnicamente como integrar aplicativos Android com a plataforma de pagamento da GETNET, assim como, requisitos de software e hardware, boas práticas de desenvolvimento e a forma de submeter o aplicativo para certificação e publicação.

Através desta API, o desenvolvedor poderá acessar as funcionalidades de Pagamento, Estorno, Pré-Autorização e Reimpressão.

Essa especificação pode ser alterada no futuro e a Getnet poderá solicitar que o aplicativo seja adaptado, principalmente com o surgimento de novos hardwares.

Kit de Desenvolvimento

Efetuar solicitação através do e-mail smartpos@getnet.com.br aos cuidados de Vinicius Pessil Bohrer..

Hardwares disponíveis

Ingenico – APOS A8



Fabricante	Ingenico
Nome	APOS A8
Processador	Principal: ARM Cortex-A7 Quad Core Criptográfico: ARM Cortex-M4 32-bit
Memória	1GB RAM+ 8GB FLASH; opcional 2GB RAM + 16GB Flash
Sistema Operacional	Android 5.1 (22) com sistema de segurança de pagamento
SIM	1 SIM
SAM	2 SAM
Interface	Tarja Magnética: ISO 1/2/3 Smart Card: EMV Level 1 Contactless: EMV LEVEL 1&2 compliant
Display	5,5"
Resolução	1280 x 720 pixels
Densidade	320 (XHDPI)
Teclado	Teclado virtual
Conectividade	WAN + LAN: 4G cat4 / 3G / 2G + WiFi bgn LAN: WiFi bgn Bluetooth: BT 3.x

Bateria	Bateria – 26000mAh x 2
GPS	Sim
Câmera	Câmera Traseira - 5M pixel com zoom automático com flash Scanner frontal dedicado para leitura de códigos 1D/2D
Impressora	Tipo – Térmica Velocidade: até 18 linhas – 58 mm largura Diâmetro da bobina - Ø 40 mm
Dimensões	183 × 84 × 64 mm
Peso	450 g
Sobre	https://www.ingenico.com.br/smart-terminals/pos/android/apos.html

Gertec – GPOS700



Fabricante	Gertec
Nome	GPOS700
Processador	QuadCore 1,2 GHz
Memória	SD RAM: 1 GB FLASH ROM: 8 GB Memória Externa: MicroSD (até 64GB)
Sistema Operacional	Android 5.1 (22)
SIM	1 SIM
SAM	2 SAM
Interface	Magnética: Bidirecional Trilhas 1, 2 e 3 - ISO 7810 e 7811 Smart Card: ISO 7816 EMV200/POBC3.0 L1/L2 Contactless / NFC: ISO 14443 Tipo A/B, Cartão Mifare® e Felica Frequência de trabalho 13.56MHz
Display	5,5"
Resolução	720 x 1280 pixels
Densidade	240 dpi (HDPI)
Teclado	Teclado virtual
Conectividade	Wi-Fi: 802.11 b/g/n com suporte a 2,4G/5G 2G: GSM/EDGE 850/ 900/ 1800/ 1900MHz 3G: WCDMA 850/900/1800/2100 MHz 4G: LTE FDD 1/3/7 e TDD 38/39/40/41 Bluetooth: 4.0
Bateria / Carregamento	Bateria - 3,7V 6000mA

	Adaptador CA - 5,5V, 2 Micro-USB
GPS	Sim
Câmera	Câmera Traseira - 5M pixel com zoom automático Código de Barras / QRCode: Leitor via câmera (1D/2D)
Impressora	Tipo - Térmica Velocidade - High speed Diâmetro da bobina - Ø 30 mm
Dimensões	206 x 84 x 57 mm
Peso	490 g
Sobre	https://www.gertec.com.br/produtos/gpos700/

Impressão

Atualmente, para efetuar impressões no equipamento, deve-se utilizar o SDK do fabricante. Contudo, no futuro, a Getnet disponibilizará o seu próprio SDK que irá abstrair os métodos de impressão para qualquer hardware.

O que você precisa saber antes de começar a desenvolver

- O desenvolvimento do aplicativo pode ser nativo ou híbrido (Exemplo: Xamarin, porque o Xamarin irá compilar o código para o “Android Runtime (ART)” como se fosse um app nativo)
- É expressamente proibido WebApps e WebView devido a medidas de segurança
- O GooglePlay Services não estará disponível
- Suporte ao Android 5.1 (22)
- É de responsabilidade do desenvolvedor garantir a segurança da informação circulada dentro do aplicativo.

Boas práticas

- Uso consciente do consumo de dados (para quando for disponibilizado o 3G no futuro)
- Uso consciente do consumo de bobina de impressão
- Uso consciente do consumo de memória e cpu
- Utilização das boas práticas de programação Android
- Evite funcionalidades com timeout infinito que possa deixar sua aplicação travada aguardando algum evento.
- Utilizar layout dinâmicos considerando que no futuro o app pode ser instalado em outros hardwares com resolução e densidade diferentes.
- Tamanho do APK
- Evite expor dados sensíveis dos clientes, tais como, senhas, informações pessoais...

Requisitos mandatórios

- O aplicativo deve possuir uma sessão de "Fale Conosco" informando ao cliente os contatos necessários para atendimento em caso de problemas ou dúvidas. Essa sessão deve estar bem visível e de fácil acesso.
- Ícone
- O nome do aplicativo não pode remeter à palavras de POS, POS DIGITAL, SMART POS, GETNET, ADQUIRENTE....
- Definição da Política de privacidade

Submetendo um Aplicativo

Até o presente momento a submissão de aplicativos se dará através do smartpos@getnet.com.br

Arquivos que devem ser enviados:

- APK Release
- Documentação
 - Fluxo de telas e informações das funcionalidades do app
 - Dados necessários para os testes (Ex.: credencias de acesso)
 - ReleaseNotes: para que a Getnet possa avaliar o escopo dos testes.
 - Quais recursos externos o aplicativo utiliza e os seus respectivos IPs/URLs (Ex. Crashlytics, Analytics, Firebase, servidores, ...).
 - Contatos para suporte ao aplicativo

Importante: Não serão aceitos aplicativos que não sejam release e com conteúdo "em construção".

Checklist mínimo da **certificação** do aplicativo feito pela Getnet, a aprovação do aplicativo não irá acontecer até que todos os itens abaixo sejam atendidos:

- **Questões de segurança**
 - WebViews - É expressamente proibido WebApps e Webviews devido a medidas de segurança
 - Acesso a recursos externos impróprios
 - Integridade dos dados
 - Quaisquer itens que a GETNET julgar uma violação de segurança
- **Requisitos Mandatórios**
- **Uso abusivo de recursos** (Bateria, Impressão, 3G)
- **Integração com a API de pagamento** - a integração deve estar de acordo com a especificação e o aplicativo não pode "quebrar".
- **Suporte ao Android 5.1 (22)**
- **Aplicativo completo Release**, as funcionalidades do app devem estar de acordo com a documentação e não podem possuir conteúdo "em construção":
 - Deve ser possível navegar por todas telas sem o aplicativo "quebrar"

API

Simulador da API

Juntamente com esta documentação, a GETNET provê um apk chamado Rebatedor.apk:

Este aplicativo simula as requisições/respostas da aplicação de pagamento, o objetivo é proporcionar para o desenvolvedor um ambiente o mais próximo possível do app Pagamento. Este simulador não irá realizar as transações online com cartões, todas as funcionalidades são "não reais" e offline - simuladas.

Integração com a API de pagamento

A integração com a API de pagamento se dará através de deeplinks parametrizados, por exemplo:

```
public class MainActivity extends AppCompatActivity {

    private final int REQUEST_CODE = 10001;

    private void startIntent(Bundle bundle, String deeplink) {
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(deeplink));
        if (bundle != null) {
            intent.putExtras(bundle);
        }
        startActivityForResult(intent, REQUEST_CODE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (RESULT_OK == resultCode && REQUEST_CODE == requestCode) {
            foo();
        }
    }
}
```

Deeplinks disponíveis

Todos os parâmetros deverão ser enviados através do Bundle do Intent e como String

URL	Função
-----	--------

<code>getnet://pagamento/v1/payment</code>	Pagamento
<code>getnet://pagamento/v1/pre-authorization</code>	Pré-autorização
<code>getnet://pagamento/v1/refund</code>	Reembolso
<code>getnet://pagamento/v1/reprint</code>	Reimpressão do último comprovante
<code>getnet://pagamento/v1/getinfos</code>	Retorna informações do terminal

Resultados

Toda funcionalidade retornará no final um resultado:

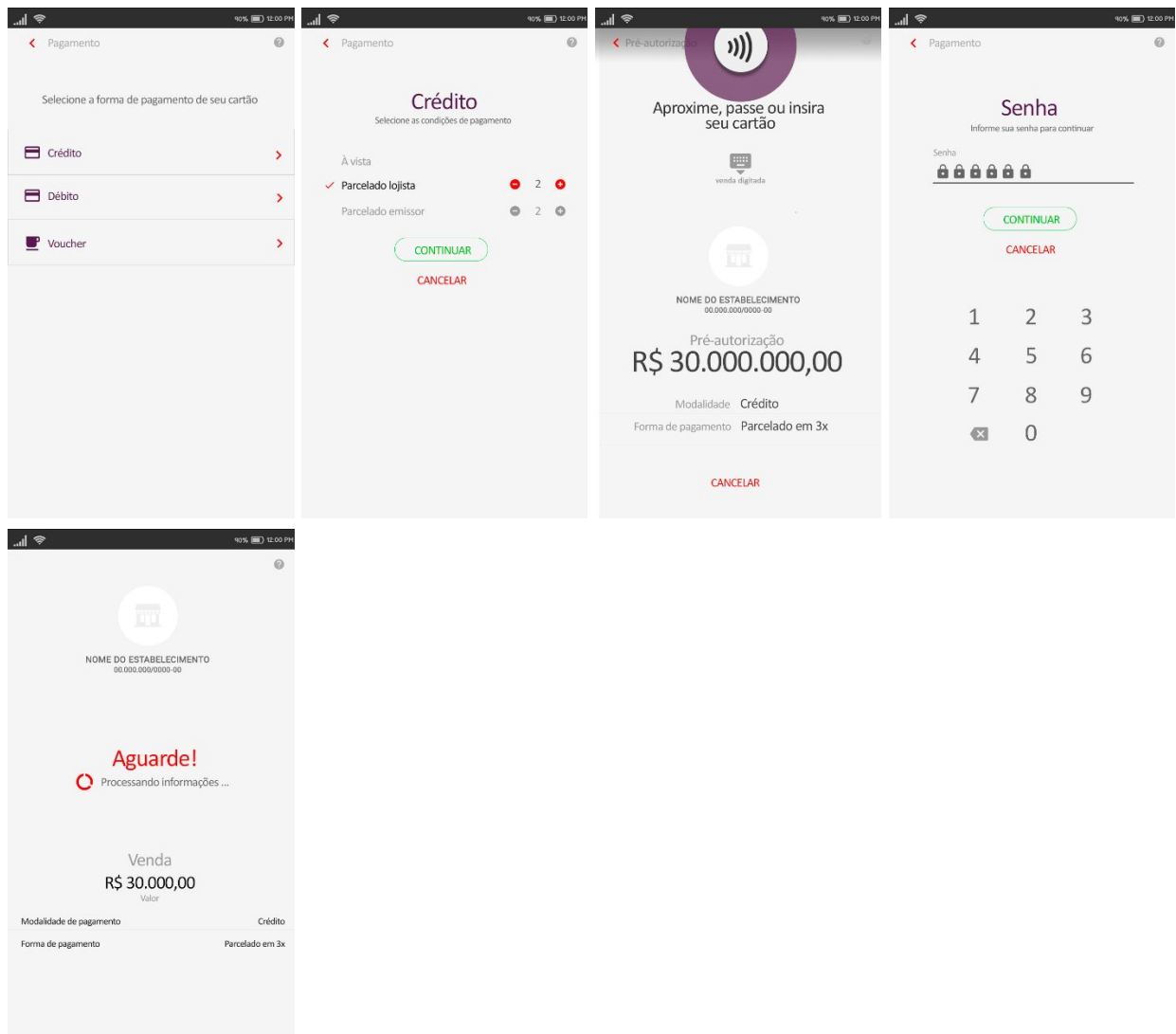
CÓDIGO	NOME	DESCRIÇÃO
0	SUCESSO	Transação efetuada com sucesso
1	NEGADA	Transação negada pelo servidor (emissor, bandeira, ...)
2	CANCELADA	Transação cancelada (pelo servidor ou usuário)
3	FALHA	Falha ao completar a transação (internet, servidor, emissor, ...)
4	DESCONHECIDO	Erro desconhecido

Funcionalidades

A seguir será especificado a API para cada funcionalidade disponível.

Pagamento

O deeplink **`getnet://pagamento/v1/payment`** irá iniciar a aplicação de pagamento da Getnet que irá tomar conta do terminal e iniciará o fluxo de pagamento solicitando primeiramente a forma de pagamento com ilustra nas imagens abaixo. **Se os dados já forem enviados no deeplink, então as telas serão automaticamente “puladas”.**



No final da transação de pagamento, se a transação for aprovada, irá apresentar a tela abaixo e a impressão da via do estabelecimento iniciará automaticamente. Ao clicar em qualquer uma das duas opções, irá apresentar a mensagem de "Remova seu cartão" e a aplicação de pagamento irá sair somente após a remoção do cartão, com exceção das vendas com tarja magnética.



Caso ocorra alguma falha na transação, irá apresentar a tela abaixo e a aplicação da Getnet irá aguardar a remoção do cartão para sair.



Abaixo seguem as tabelas de requisição e resposta e seus respectivos parâmetros.

Requisição		
	Parâmetro	Descrição
OPCIONAL	paymentType	<p>Este parâmetro informa qual o tipo de pagamento que será feito:</p> <p>"credit", "debit", "voucher"</p> <p>Se esse campo for informado a tela de seleção de tipo de pagamento será pulada.</p>
OPCIONAL	creditType	<p>Se informado no parâmetro "paymentType" crédito então este parâmetro deverá informar qual tipo de crédito:</p> <p>"creditMerchant" - Crédito parcelado Lojista "creditIssuer" - Crédito parcelado Emissor</p> <p>Se selecionado crédito e este campo for informado então a tela de tipo e crédito será pulada. <u>Crédito à vista não precisa enviar este campo.</u></p>
OPCIONAL	installments	<p>Este parâmetro informa o número de parcelas.</p> <p>Se informado o parâmetro "creditType" com o valor "creditMerchant" ou "creditIssuer" então este parâmetro deve ser informado.</p>
OPCIONAL	amount	<p>12 dígitos representando o valor, considerando os últimos 2 dígitos como casas decimais. Exemplo: "000000001234" é equivalente R\$ 12,34</p> <p>Se não informado o usuário precisará preencher o valor no app de Pagamento</p>
OPCIONAL	currencyPosition	<p>"CURRENCY_AFTER_AMOUNT" ou "CURRENCY_BEFORE_AMOUNT"</p> <p>Default : "CURRENCY_BEFORE_AMOUNT"</p>
OPCIONAL	currencyCode	<p>Código da moeda, de acordo com a ISO-4217 Exemplo: "986" este é código do Real (R\$) Para a lista completa das moedas acesse: https://pt.wikipedia.org/wiki/ISO_4217</p>

		Default: "986"
Resposta		
Quando retorna?	Parâmetro	Descrição
SEMPRE	result	Resultado da transação - para mais informações consulte a legenda deste campo
OPCIONAL	resultDetails	Texto com detalhes do retorno
SEMPRE	amount	12 dígitos representando o valor, considerando os últimos 2 dígitos como casas decimais. Exemplo: 000000001234 = R\$ 12,34
OPCIONAL	nsu	Código de autorização da transação
OPCIONAL	cvNumber	Número do CV
SEMPRE	type	02 - Débito 11 - Crédito a vista 12 - Crédito parcelado Lojista 13 - Crédito parcelado Emissor 03 - Voucher
OPCIONAL	brand	Bandeira do cartão utilizado
SEMPRE	inputType	021 - tarja magnética 051 - chip 071 - chip sem contato 801 - tarja magnética - fallback
OPCIONAL	installments	Quantidade de parcelas selecionada

Exemplo de implementação

```
public class MainActivity extends AppCompatActivity {

    private final int REQUEST_CODE = 1001;
    private final String ARG_RESULT = "result";
    private final String ARG_RESULT_DETAILS = "resultDetails";
    private final String ARG_AMOUNT = "amount";
    private final String ARG_TYPE = "type";
    private final String ARG_INPUT_TYPE = "inputType";
    private final String ARG_INSTALLMENTS = "installments";
    private final String ARG_NSU = "nsu";
    private final String ARG_BRAND = "brand";

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Bundle bundle = new Bundle();
        bundle.putString("amount", "000000001000");
        bundle.putString("currencyPosition", "CURRENCY_AFTER_AMOUNT");
        bundle.putString("currencyCode", "986");
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("getnet://pagamento/v1/payment"));
        intent.putExtras(bundle);
        startActivityForResult(intent, REQUEST_CODE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (REQUEST_CODE == requestCode && RESULT_OK == resultCode) {
            String result = data.getString(ARG_RESULT);
            String resultDetails = data.getString(ARG_RESULT_DETAILS);
            String amount = data.getString(ARG_AMOUNT);
            String type = data.getString(ARG_TYPE);
            String inputType = data.getString(ARG_INPUT_TYPE);
            String installments = data.getString(ARG_INSTALLMENTS);
            String nsu = data.getString(ARG_NSU);
            String brand = data.getString(ARG_BRAND);
        }
    }
}
```

Pré-autorização

O deeplink **getnet://pagamento/v1/pre-authorization** irá iniciar o app de Pagamento na tela de Pré-autorização. Segue as tabelas de requisição/resposta e seus respectivos parâmetros.

Requisição		
	Parâmetro	Descrição
MANDATÓRIO	amount	12 dígitos representando o valor, considerando os últimos 2 dígitos como casas decimais. Exemplo: "000000001234" é equivalente R\$ 12,34
MANDATÓRIO	currencyPosition	"CURRENCY_AFTER_AMOUNT" ou "CURRENCY_BEFORE_AMOUNT"
MANDATÓRIO	currencyCode	Código da moeda, de acordo com a ISO-4217 Exemplo: "986" este é código do Real (R\$) Para a lista completa das moedas acesse: https://pt.wikipedia.org/wiki/ISO_4217
Resposta		
Quando retorna?	Parâmetro	Descrição
SEMPRE	result	Resultado da transação - para mais informações consulte a legenda deste campo
OPCIONAL	resultDetails	Texto com detalhes do retorno
SEMPRE	amount	12 dígitos representando o valor, considerando os últimos 2 dígitos como casas decimais. Exemplo: 000000001234 = R\$ 12,34
OPCIONAL	nsu	Código de autorização da transação
SEMPRE	type	11 - Crédito a vista 12 - Crédito parcelado Lojista
OPCIONAL	brand	Bandeira do cartão utilizado
SEMPRE	inputType	021 - tarja magnética 051 - chip 071 - chip sem contato 801 - tarja magnética - fallback
OPCIONAL	installments	Quantidade de parcelas selecionada

Exemplo de implementação

```
public class MainActivity extends AppCompatActivity {

    private final int REQUEST_CODE = 1001;
    private final String ARG_RESULT = "result";
    private final String ARG_RESULT_DETAILS = "resultDetails";
    private final String ARG_AMOUNT = "amount";
    private final String ARG_TYPE = "type";
    private final String ARG_INPUT_TYPE = "inputType";
    private final String ARG_INSTALLMENTS = "installments";
    private final String ARG_NSU = "nsu";
    private final String ARG_BRAND = "brand";

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Bundle bundle = new Bundle();
        bundle.putString("amount", "000000001000");
        bundle.putString("currencyPosition", "CURRENCY_AFTER_AMOUNT");
        bundle.putString("currencyCode", "986");
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("getnet://pagamento/v1/pre-authorization"));
        intent.putExtras(bundle);
        startActivityForResult(intent, REQUEST_CODE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (REQUEST_CODE == requestCode && RESULT_OK == resultCode) {
            String result = data.getString(ARG_RESULT);
            String resultDetails = data.getString(ARG_RESULT_DETAILS);
            String amount = data.getString(ARG_AMOUNT);
            String type= data.getString(ARG_TYPE);
            String inputType= data.getString(ARG_INPUT_TYPE);
            String installments= data.getString(ARG_INSTALLMENTS);
            String nsu= data.getString(ARG_NSU);
            String brand= data.getString(ARG_BRAND);
        }
    }
}
```

Estorno

O deeplink **getnet://pagamento/v1/refund** irá iniciar o app de Pagamento na tela de estorno. Segue as tabelas de resposta e seus respectivos parâmetros.

Se a requisição for enviada com algum campo informado abaixo, o valor dele será preenchido automaticamente, se todos os campos forem enviados, a próxima tela a ser mostrada será a de inserir o cartão.

Requisição		
	Parâmetro	Descrição
OPCIONAL	amount	12 dígitos representando o valor, considerando os últimos 2 dígitos como casas decimais. Exemplo: 000000001234 = R\$ 12,34
OPCIONAL	transactionDate	Data da transação a ser estornada. Por padrão será o dia corrente. Enviar no formato: "dd/MM/yyyy"
OPCIONAL	cvNumber	Número do CV da transação a ser estornada
OPCIONAL	originTerminal	Número lógico do terminal que efetuou a transação a ser estornada. Se não for informado, será utilizado o número lógico do terminal que está executando o procedimento.

Resposta		
Quando retorna?	Parâmetro	Descrição
SEMPRE	result	Resultado da transação - para mais informações consulte a legenda deste campo
OPCIONAL	resultDetails	Texto com detalhes do retorno
SEMPRE	amount	12 dígitos representando o valor, considerando os últimos 2 dígitos como casas decimais. Exemplo: 000000001234 = R\$ 12,34

--	--	--

Exemplo de implementação

```
public class MainActivity extends AppCompatActivity {

    private final int REQUEST_CODE = 1001;
    private final String ARG_RESULT = "result";
    private final String ARG_RESULT_DETAILS = "resultDetails";
    private final String ARG_AMOUNT = "amount";

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("getnet://pagamento/v1/refund"));
        startActivityForResult(intent, REQUEST_CODE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (REQUEST_CODE == requestCode && RESULT_OK == resultCode) {
            String result = data.getString(ARG_RESULT);
            String resultDetails = data.getString(ARG_RESULT_DETAILS);
            String amount = data.getString(ARG_AMOUNT);
        }
    }
}
```

Reimpressão

O deeplink **getnet://pagamento/v1/reprint** irá reimprimir o último comprovante. Segue a tabela com os parâmetros de resposta

Response		
Quando retorna?	Chave	Descrição
SEMPRE	result	Resultado da transação - para mais informações consulte a legenda deste campo

Exemplo de implementação

```
public class MainActivity extends AppCompatActivity {

    private final int REQUEST_CODE = 1001;
    private final String ARG_RESULT = "result";

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("getnet://pagamento/v1/reprint"));
        startActivityForResult(intent, REQUEST_CODE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (REQUEST_CODE == requestCode && RESULT_OK == resultCode) {
            String result = data.getString(ARG_RESULT);
        }
    }
}
```

Info

O deeplink **getnet://pagamento/v1/getinfos** irá retornar os dados do terminal via `activityResult`.
 Segue a tabela com os parâmetros de resposta

Resposta		
Quando retorna?	Chave	Descrição
SEMPRE	result	Resultado da transação - para mais informações consulte a legenda deste campo
SEMPRE	ec	Número do estabelecimento comercial vinculado
SEMPRE	numserie	Número de série do equipamento
SEMPRE	numlogic	Número lógico do equipamento vinculado
SEMPRE	version	Versão da API

SEMPRE	cnpjEC	CNPJ do estabelecimento vinculado à Getnet
SEMPRE	nomeEC	Nome do estabelecimento vinculado

Exemplo de implementação

```
public class MainActivity extends AppCompatActivity {

    private final int REQUEST_CODE = 1001;
    private final String ARG_RESULT = "result";
    private final String ARG_EC = "ec";
    private final String ARG_NUM_SERIE = "numserie";
    private final String ARG_NUM_LOGIC = "numlogic";
    private final String ARG_VERSION = "version";
    private final String ARG_CNPJ = "cnpjEC";
    private final String ARG_NAME = "nomeEC";

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("getnet://pagamento/v1/getinfos"));
        startActivityForResult(intent, REQUEST_CODE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (REQUEST_CODE == requestCode && RESULT_OK == resultCode) {
            String result = data.getString(ARG_RESULT);
            String ec = data.getString(ARG_EC);
            String numSerie = data.getString(ARG_NUM_SERIE);
            String numLogic = data.getString(ARG_NUM_LOGIC);
            String apiVersion = data.getString(ARG_VERSION);
            String cnpj = data.getString(ARG_CNPJ);
            String name = data.getString(ARG_NAME);
        }
    }
}
```

Dúvidas Frequentes

1. A Getnet possui um SDK para abstração de hardware?

Ainda não. Este SDK está em construção, e no futuro será disponibilizado. Por enquanto, para usar os recursos do hardware, deve-se verificar diretamente com o fabricante por intermédio da Getnet.

2. Quem deve imprimir o comprovante da transação de pagamento?

A aplicação de pagamento da Getnet. Quando o seu aplicativo chamar o método "StartActivityForResult", a aplicação da Getnet tomará conta da tela do terminal e executará toda transação de pagamento até a impressão do CV e remoção do cartão (em caso de chip). Inclusive mostrará a mensagem na tela de sucesso ou erro. Essa mensagem será retornada para seu aplicativo com resposta da API.

3. A Getnet pode me enviar o apk de pagamento para eu fazer testes?

Não. Por questões de segurança, a aplicação de pagamento da Getnet não pode ser enviada. Por isso criamos o simulador, com ele você pode simular o protocolo e as chamadas da API.

4. Como vou saber se a integração do meu aplicativo com a aplicação da Getnet está funcionando?

A utilização do simulador garante que a API está de acordo, porém somente durante a certificação interna da Getnet que teremos a validação completa. Durante esta certificação, a Getnet utiliza a aplicação de pagamento oficial para os testes de integração.

5. Toda vez que eu mandar uma versão nova a Getnet irá certificar?

Depende. Através do releaseNotes que está na documentação, a Getnet irá avaliar o escopo dos testes e se será necessário executá-los. Contudo, sempre serão executadas as ferramentas de testes de segurança.

6. Meu aplicativo ainda não está pronto, mas eu posso iniciar a certificação da Getnet para avaliar a integração com pagamento?

Não, pois a Getnet entende que um aplicativo submetido já está validado pelo desenvolvedor e apto para ser publicado para produção. Inclusive, se o apk for aprovado pela certificação, o mesmo será assinado pela Getnet e já estará pronto para produção.

7. Todo o controle da transação de adquirencia será realizado pela aplicação de pagamento da Getnet? Inclusive a situação das transações pendentes e confirmadas?

Sim, a aplicação de pagamento é responsável por toda parte da adquirencia.

8. Eu terei que colocar o meu simcard 3G no POS?

Não, o simcard será da Getnet.

9. Existe alguma limitação ou requisito para o uso do Wifi?

Sim, o POS Digital, por questões de segurança e PCI, não pode se conectar à uma rede wifi sem segurança (sem senha), assim como também não pode entrar em redes que exijam autenticação via browser.

10. Eu gostaria de fazer a leitura de cartões mifare no meu aplicativo, é possível?

Sim, atualmente, nos terminais da Gertec (GPOS700), você pode utilizar o SDK nativo do Android. Para o terminal da Ingenico (APOS A8), é necessário utilizar o sdk da Ingenico. Futuramente, a Getnet irá disponibilizar um SDK que facilite este recurso.

11. Qual o contato da Ingenico?

BRGDAPOS@ingenico.com

12. Qual o contato da Gertec?

Antonio Eduardo Campos Falcao <eduardo.falcao@gertec.com.br>

13. A versão do android é 5.1 (Lollipop), basta minha aplicação suportar o Android 5.1 (22) ou necessariamente tenho que compilar para SDK 22?

Isso. Basta ter suporte para versão 22.

14. Posso compilar em um SDK superior e informar a minSDK 22?

O target pode ser superior, mas o inSDK deve ser 22.

