



ESPECIFICAÇÃO WEBSERVICES

V4.4.12

Toda a informação constante neste documento é confidencial. Qualquer entidade com acesso a este documento deve manter a sua confidencialidade, sem redistribuir parcial ou integralmente o seu conteúdo.

Luís Oliveira/José Ribeiro/Márcio Monteiro/Hugo Ferreira
01/03/2016

Conteúdo

1. Introdução e objetivos	5
2. Sugestões de implementação dos serviços LUSOPAY	6
2.1. Referências multibanco e payshop	6
2.1.1. Pagamentos em lojas online	6
2.1.2. Sistemas de faturação ERP	6
2.2. Contas de pagamento LUSOPAY	8
2.2.1. Pagamentos em lojas online	8
2.2.2. Sistemas de faturação (ERP).....	9
2.2.3. Pagamentos em software POS	10
2.2.4. Pagamentos em bombas de combustível com atendimento automático (self-service) ou outros serviços que exijam «One Time Password» e a função de cativar saldo antes de se saber qual o montante da compra	11
2.3. Serviço MBWay	13
3. Arquitetura	15
4. Métodos wallet.....	16
4.1. sendPaymentRequest.....	17
4.2. cancelPaymentRequest	18
4.3. listPaymentRequests	18
4.4. paymentRequestDetails	19
4.5. checkUserVatNumber	20
4.6. getUserStatus.....	21
4.7. addAwaitingPayment	22
4.8. getAwaitingPayment	22
4.9. cancelAwaitingPayment	23
4.10. commitAwaitingPayment.....	24
4.11. getAwaitingPaymentStatus.....	25
4.12. preparePaymentTransaction.....	25
4.13. initializePaymentTransaction	26
4.14. commitPaymentTransaction	28
4.15. cancelPaymentTransaction	30
5. Métodos serviços de referências	31
5.1. getMovementsByUser	31
5.2. getNewReference	33
5.3. getNewMovements	34
5.4. getPaymentDate	35

5.5.	getKey	36
5.6.	getNewDynamicReference.....	37
5.7.	getNewDynamicMovements.....	38
5.8.	getNewDynamicMovementsByDates	39
5.9.	getCheckDigitByReference	40
5.10.	getNewReferencePayshopNet	41
5.11.	getBalanceByUser	41
6.	Métodos públicos	43
6.1.	getBankList.....	43
7.	Objectos	44
7.1.	MovementType	44
7.2.	References	44
7.3.	ReferenceType	44
7.4.	ReferencesGenerationFeedback	44
7.5.	SendPaymentRequestFeedback.....	44
7.6.	PaymentRequestStatus	45
7.7.	PreparePaymentTransactionFeedback.....	45
7.8.	InitializePaymentTransaction.....	45
8.	Serviços MBWay.....	45
8.1.	sendMBWayRequest.....	45
	ANEXOS	47

Histórico de alterações

Versão	Data	Autor	Alterações
V4.0	2015-03-09	Luís Oliveira	<ul style="list-style-type: none"> • Conversão do serviço para utilizar a tecnologia WCF; • Uniformização dos parâmetros de entrada e saída; • Adição de parâmetros de retorno, conforme sugestões e necessidades;
V4.1	2015-04-09	Luís Oliveira	<ul style="list-style-type: none"> • Inclusão do método <i>getUserStatus</i>;
V4.2	2015-04-30	Luís Oliveira	<ul style="list-style-type: none"> • Inclusão do método <i>cancelPaymentRequest</i>; • Inclusão do método <i>listPaymentRequests</i>; • Inclusão do método <i>checkUserVatNumber</i>;
V4.3	2015-05-05	Luís Oliveira	<ul style="list-style-type: none"> • Inclusão do método <i>paymentRequestDetails</i>; • Inclusão do método <i>addAwaitingPayment</i>; • Inclusão do método <i>getAwaitingPayment</i>;
V4.4	2015-05-11	Luís Oliveira	<ul style="list-style-type: none"> • Inclusão do método <i>preparePaymentTransaction</i>; • Inclusão do método <i>initializePaymentTransaction</i>; • Inclusão do método <i>commitPaymentTransaction</i>; • Inclusão do método <i>cancelPaymentTransaction</i>;
V4.4.1	2015-05-14	Luís Oliveira	<ul style="list-style-type: none"> • Colocação da indicação que nos métodos <i>initializePaymentTransaction</i>, <i>commitPaymentTransaction</i> e <i>cancelPaymentTransaction</i>, o parâmetro <i>cellPhoneNumber</i> deve incluir o indicativo do país;
V4.4.2	2015-05-15	Luís Oliveira	<ul style="list-style-type: none"> • Alteração do valor mínimo para geração de referências de 2€ para 1.2€;
V4.4.3	2015-06-03	Luís Oliveira	<ul style="list-style-type: none"> • Inclusão no documento, da mensagem de erro lançada pelo serviço, quando o método <i>getNewDynamicReference</i> é invocado com um pedido de geração de mais de 25 referências.
V4.4.4	2015-07-01	José Ribeiro	<ul style="list-style-type: none"> • Inclusão no documento de explicações e sugestões de utilização dos métodos constantes neste manual

			<ul style="list-style-type: none"> • Alteração do ponto 1 – Introdução e objetivos • Criação de um novo ponto: 2 • Retificação dos anteriores pontos 2, 3 e 4 para 3, 4 e 5 • Atualização do índice
V4.4.5	2015-08-27	Luís Oliveira	<ul style="list-style-type: none"> • Organização dos métodos em capítulos, de acordo com o seu âmbito de operação, nomeadamente, serviços de referências, wallet ou gerais;
V4.4.6	2015-09-04	Márcio Monteiro	<ul style="list-style-type: none"> • Inclusão de estrutura de QRCode em anexo
V4.4.7	2015-09-22	Luís Oliveira	<ul style="list-style-type: none"> • Alteração do método <i>addWaitingPayment</i>, adicionando-lhe o parâmetro <i>externalReference</i>; • Inclusão do método do método <i>cancelAwaitingPayment</i>; • Inclusão do método do método <i>commitAwaitingPayment</i>; • Inclusão do método do método <i>getAwaitingPaymentStatus</i>;
V4.4.8	2015-09-29	José Ribeiro	<ul style="list-style-type: none"> • Alteração da estrutura do QR Code
V4.4.9	2015-10-05	José Ribeiro	<ul style="list-style-type: none"> • Alteração da estrutura do QR Code
V4.4.10	2016-01-11	Márcio Monteiro	<ul style="list-style-type: none"> • Inclusão de link de acesso a exemplos de programação e manual de callback nos anexos
V4.4.11	2016-01-20	Hugo Ferreira	<ul style="list-style-type: none"> • Inclusão do método <i>getBalanceByUser</i>; • Alteração do método <i>getMovementsByUser</i>; para mostrar mais do que uma conta se existir.
V4.4.12	2016-03-01	Hugo Ferreira	<ul style="list-style-type: none"> • Novo serviço MBWay • Inclusão do método <i>sendMBWayRequest</i>

1. Introdução e objetivos

A LUSOPAY disponibiliza aos seus clientes, a contratação do serviço Multibanco e Payshop, que permite disponibilizar a geração de referências de pagamento na web e a sua integração nos mais diversos sistemas de informação, como por exemplo, lojas virtuais, ERPs, apps, entre outros.

Para além disso, disponibiliza um serviço de contas de pagamento, que possibilita aos clientes LUSOPAY, receberem fundos nas suas contas LUSOPAY, enviarem fundos para contas bancárias e transacionarem fundos entre utilizadores da rede LUSOPAY, tudo isso à distância de um toque no telemóvel ou pela utilização do «homebanking» da LUSOPAY. Isto porque as contas de pagamento LUSOPAY tanto podem ser acedidas por página web, como por apps (aplicações móveis).

O objetivo deste documento é apresentar e descrever, os serviços que a LUSOPAY disponibiliza, para que seja possível a integração destes em sistemas informáticos de terceiros ou *open source*, tanto por geração de referências de pagamento Multibanco e Payshop, como através da utilização das contas de pagamento da LUSOPAY.

Assim, neste documento serão apresentados em detalhe os diversos métodos que compõem o webservice de pagamentos da LUSOPAY.

2. Sugestões de implementação dos serviços LUSOPAY

2.1. Referências multibanco e payshop

2.1.1. Pagamentos em lojas online

Pressupostos:

No módulo dedicado ao método de pagamento por referências multibanco e payshop, deve existir um espaço para identificação da empresa dona da loja online.

Essa identificação faz-se por: NIF e ClientGuID

Lógica de funcionamento:

No processo de finalização da compra, existe um momento onde o cliente decide qual a forma de pagar que prefere dentro das possíveis nessa loja online.

O cliente seleciona a opção Referências Multibanco / Payshop.

Aparecerá uma janela com as referências multibanco e payshop para o cliente utilizar no pagamento através da invocação do método

GetNewDynamicReference.

Assim que o cliente efetue o pagamento da referência, seja num homebanking ou num ATM, a loja online pode receber essa informação, atualizando o estado da encomenda para Pagamento Confirmado ou Pago. Para tal, deve implementar o exposto no nosso Manual de ativação da funcionalidade confirmação automática de pagamento (manual de callback).

Pode igualmente dar baixa dos stocks de produtos de forma automática.

Poderá consultar um vídeo no YouTube do desenvolvimento de um módulo de referências multibanco para lojas online em plataforma Prestashop:

<https://www.youtube.com/watch?v=V-r8jKVbgLk>

2.1.2. Sistemas de faturação ERP

O software de faturação ou ERP deve ter na ficha da empresa a quem o software está licenciado, ou no módulo de referências multibanco/payshop um espaço para identificação da conta LUSOPAY dessa empresa. Esse identificador é o ClientGuID e o NIF.

Poderá configurar nas fichas de cliente dos clientes da empresa que utiliza o software licenciado, uma indicação se se pretende ou não gerar referências multibanco ou payshop para aquele cliente em concreto. Pode haver casos, em que a empresa saiba que aquele cliente só paga por multibanco ou por payshop.

Para empresas com uma lista grande de clientes, será aconselhável a possibilidade de atribuir em lote a possibilidade de gerar referências multibanco e/ou payshop para toda a lista de clientes.

Lógica de funcionamento:

Vamos explicar uma filosofia de implementação que implica acesso à Internet e outra que não exige acesso à Internet por parte do software de faturação.

Exemplo de desenvolvimento que implica acesso à Internet por parte do software de faturação/ERP

Ao emitir uma fatura isolada ou ao processar um conjunto de avenças, se o cliente destinatário da fatura for cliente sujeito à geração de referências multibanco e /ou payshop de acordo com o definido na ficha de cliente, o software de faturação/ERP pedirá ao webservice da LUSOPAY a geração de referências através da invocação do método **GetNewDynamicReference**. Estas são devolvidas pelo web service e impressas na fatura.

O cliente recebendo a fatura com as referências multibanco e/ou payshop, poderá pagar por essa via.

Para receber informação sobre pedidos de pagamentos aceites, deverá consultar o Manual de ativação da funcionalidade confirmação automática de pagamento.

Exemplo de desenvolvimento que NÃO implica acesso à Internet por parte do software de faturação/ERP

Neste caso, em vez de se solicitar a um web service a devolução de uma referência multibanco ou payshop, o cliente utilizador do serviço contrata com a LUSOPAY um conjunto de referências a que chamamos Pacotes de referências Multibanco ou Pacotes de referências Payshop.

Estes pacotes podem ser de 20, 100, 1000 ou 10.000 referências e seus múltiplos.

Ex: um cliente pode querer reservar 60 referências. Compra 3 pacotes de 20 referências.

No desenvolvimento de software que use pacotes de referências, é boa política permitir que o cliente possa adicionar mais pacotes.

É necessário incluir na programação o algoritmo para a geração de referências multibanco e payshop.

Ao imprimir faturas, ou outros documentos, tais como cartas de cobrança para clientes relapsos, deverá ser possível incluir as referências multibanco e payshop.

O software deve guardar as referências criadas e a associação destas às faturas ou conjunto de faturas (no caso de cartas de cobrança que juntem mais do que uma fatura e que incluam uma referência para liquidação).

A área de clientes da LUSOPAY permite que os nossos clientes possam ver quais as referências liquidadas.

Não tendo possibilidade de ligação à Internet, o software deve permitir que o cliente pesquise as referências anteriormente criadas para indicar ao software que determinada referência foi paga, permitindo a emissão de recibo ou na indicação de que essa fatura foi liquidada.

2.2. Contas de pagamento LUSOPAY

2.2.1. Pagamentos em lojas online

Pressupostos:

No módulo dedicado ao método de pagamento LUSOPAY, deve existir um espaço para identificação da empresa dona da loja online.

A LUSOPAY solicita as seguintes informações de configuração: ClientGUID e NIF.

Lógica de funcionamento:

No processo de finalização da compra, existe um momento onde o cliente decide qual a forma de pagar que prefere dentro das possíveis nessa loja online.

O cliente seleciona a opção LUSOPAY Wallet.

Aparecerá uma janela onde a plataforma de e-commerce questiona o cliente sobre qual o seu nome de utilizador LUSOPAY. Recomendamos vivamente a existência de um segundo campo para validação do nome de utilizador (para evitar erros de digitação) e uma informação de quais os passos a efetuar após a finalização da encomenda.

Exemplo:

Digite o seu nome de utilizador LUSOPAY:

Confirme o seu nome de utilizador LUSOPAY:

Após clicar no botão enviar, vá a www.lusopay.com e entre na sua conta LUSOPAY com o seu nome de utilizador e senha. Em notificações, encontrará um pedido de pagamento desta sua encomenda. Aceite e a encomenda ser-lhe-á enviada.

Note que, ultrapassado o prazo máximo para aceitação do pagamento, esta encomenda será anulada.

Enviar

Após clicar no botão enviar, é validado se o nome existente no primeiro campo é igual ao do segundo.

Se for diferente, alerta para essa diferença e pede retificação.

Se for igual, é enviado um pedido de pagamento para o webservice da LUSOPAY, contendo os dados especificados no método **sendPaymentRequest**

Assim que o cliente confirme no homebanking ou app LUSOPAY o pedido de pagamento, a transação é efetuada, recebendo a loja online essa informação, atualizando o estado da encomenda para Pagamento Confirmado ou Pago. Pode igualmente dar baixa dos stocks de produtos de forma automática.

Para receber informação sobre pagamentos recebidos, deverá consultar o Manual Callback da LUSOPAY.

2.2.2. Sistemas de faturação (ERP)

Pressupostos:

O software de faturação ou ERP deve ter na ficha da empresa a quem o software está licenciado um espaço para identificação da conta LUSOPAY dessa empresa. Esse identificador é o ClientGuID e NIF.

Nas fichas de clientes da empresa também deve existir um campo para especificar o nome de utilizador LUSOPAY da empresa ou particular que é cliente da empresa utilizadora do software.

Lógica de funcionamento:

Vamos explicar uma filosofia de implementação que implica acesso à Internet e outra que não exige acesso à Internet por parte do software de faturação.

Exemplo que desenvolvimento que implica acesso à Internet por parte do software de faturação/ERP

Ao emitir uma fatura isolada ou ao processar um conjunto de avenças, se o cliente destinatário da fatura for cliente LUSOPAY e o seu nome de utilizador estiver registado na ficha de cliente, o software de faturação/ERP envia automaticamente o(s) pedido(s) de pagamento para o webservice da LUSOPAY. Para enviar pedidos de pagamento, use o método **sendPaymentRequest**.

O cliente recebe no seu homebanking esse pedido de pagamento, podendo aceitar ou recusar.

Se o software de faturação/ERP emitir os pedidos de pagamento, e permitir outras formas de pagamento, tais como referências multibanco, payshop, transferências bancárias, sempre que der como paga uma fatura que não tenha sido liquidada pelo pedido de pagamento enviado para o homebanking da LUSOPAY, o software deve cancelar o pedido de pagamento através do método **cancelPaymentRequest**. Assim, evitam-se situações de pagamentos em duplicado da mesma fatura.

Para receber informação sobre pedidos de pagamentos LUSOPAY Wallet aceites, deverá invocar um dos seguintes método: `listPaymentRequests`, para obter a listagem dos pedidos e `paymentRequestDetails`, para obter o detalhe de cada um.

Se o software de faturação/ERP tiver implementado quer o serviço LUSOPAY Wallet, quer o serviço de referências multibanco e payshop, para saber se uma referência multibanco ou payshop foi liquidada, deverá implementar o sistema seguindo o Manual Callback da LUSOPAY.

Exemplo de desenvolvimento que NÃO implica acesso à Internet por parte do software de faturação/ERP

Neste caso, o pedido de pagamento será feito na própria fatura, imprimindo na fatura um espaço com o logo da LUSOPAY e o nome de utilizador da empresa que emitiu a fatura. Sugerimos que neste caso, se peça ao pagador para especificar o número da fatura no descritivo do pagamento.

Exemplo:



Pague esta fatura por LUSOPAY

Nosso nome de utilizador:

XXXXXXXXXXXXXXXXXXXX

Agradecemos que no descritivo do pagamento especifique o número da fatura.

Outra solução, pode ser a impressão de um QRCode ou código de barras que identifica a empresa emitente da fatura. O pagador pode utilizar a sua app LUSOPAY para ler o QR Code ou código de barras. Ao ler o QR Code ou código de barras, a app preenche automaticamente os campos do destinatário do pagamento, tendo o pagador apenas que especificar o valor e descrição. Como neste caso não há acesso à Internet, a emissão de recibos terá de ser manual por análise do extrato de conta LUSOPAY, tal como acontece com o extrato bancário para as tradicionais transferências bancárias.

2.2.3. Pagamentos em software POS **Pressupostos**

O software de faturação ou ERP deve ter na ficha da empresa a quem o software está licenciado um espaço para identificação da conta LUSOPAY dessa empresa. Esse identificador é o ClientGUID e NIF.

Lógica de funcionamento:

Estando o cliente no caixa para pagar e concluída a conta da compra, o cliente refere que quer pagar por LUSOPAY Wallet.

O operador de caixa carrega num botão (num touchscreen) onde se especifica qual a forma de pagamento.

Ao clicar nesse botão que diz que o cliente quer pagar por LUSOPAY, é enviado um pedido de pagamento para o webservice da LUSOPAY. Para enviar o pedido de pagamento, usa-se o método **addAwaitingPayment**.

Após este envio, o caixa informa o pagador de que este pode ler o QR Code LUSOPAY junto à caixa registadora.

Esta leitura de QR Code faz com que a app LUSOPAY baixe o último pedido de pagamento existente nos servidores da LUSOPAY referentes àquela caixa registadora em concreto, invocando o método **getAwaitingPayment**.

Uma vez recebido o resumo da conta, o cliente analisa e clica num botão para aceitar ou recusar a conta.

Assim que o pagador clicar no botão «Aceitar», o estado do pagamento em espera é atualizado para “Pago”, através da invocação do método **commitAwaitingPayment**.

Caso o comerciante deseje cancelar a espera do pagamento deve ser invocado o método **cancelAwaitingPayment**.

Para obter a informação do pedido de pagamento o POS deverá invocar periodicamente o método **getAwaitingPaymentStatus**.

Quando o operador da caixa verificar que o valor foi pago na totalidade, o que poderá acontecer por vários pagamentos parciais, poderá avançar para a impressão do recibo, indicando forma de pagamento LUSOPAY.

2.2.4. Pagamentos em bombas de combustível com atendimento automático (self-service) ou outros serviços que exijam «One Time Password» e a função de cativar saldo antes de se saber qual o montante da compra

Agora, queremos explicar o processo de reserva prévia de saldo da conta de pagamento, feito antes da própria compra.

Aplica-se quando não se sabe qual o valor que irá ser consumido

Procedimento

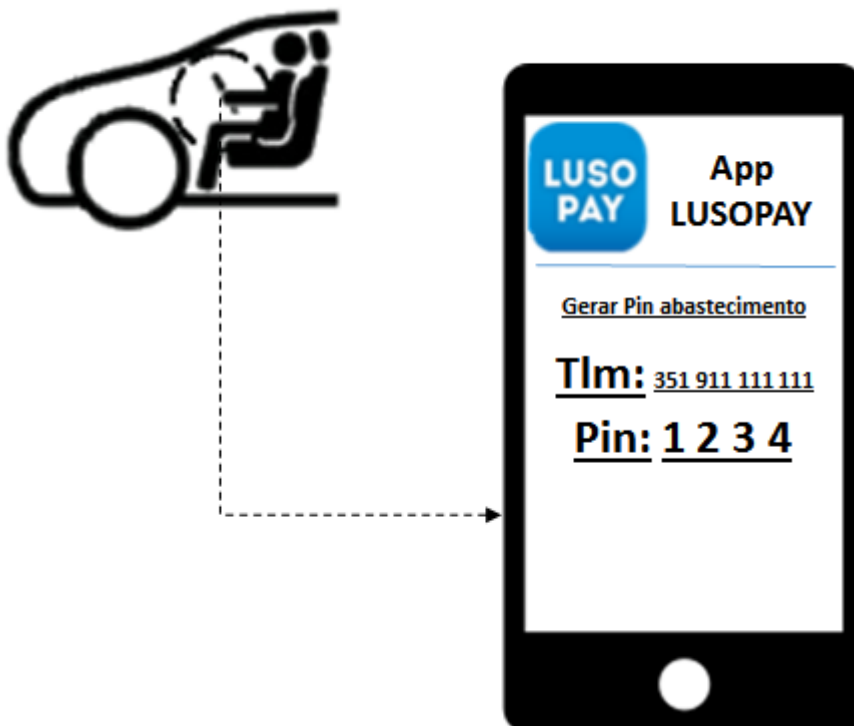
O cliente aproxima a viatura de um posto de abastecimento com suporte LUSOPAY, como meio de pagamento



Deve ser disponibilizado ao cliente uma forma de selecionar, no posto de abastecimento, a LUSOPAY como operador de pagamento.

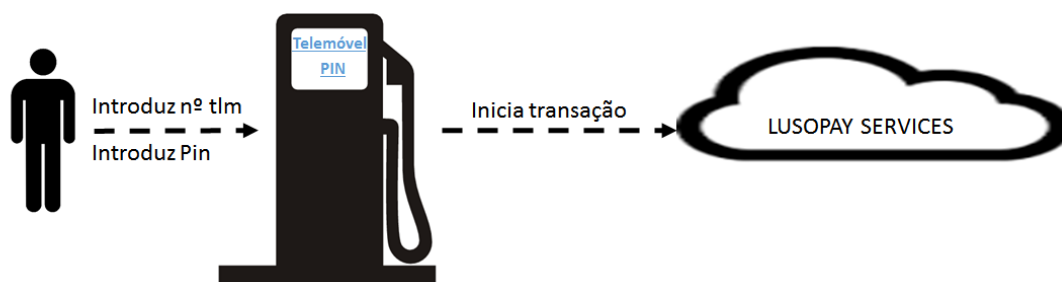
Geração do código para autenticação da transação

Antes do cliente sair da viatura, deve gerar através da app LUSOPAY, um PIN para autenticar a



transação.

Início da transação



Depois de selecionar a LUSOPAY como operador de pagamento, vamos necessitar de dar início à transação de reserva e pagamento, tendo para tal de identificar as partes envolvidas na transação.

O broker será identificado por um token (GUID), que será fornecido pela LUSOPAY.

Esse token não deverá ser difundido uma vez que é com ele que a LUSOPAY irá validar as permissões de acesso aos seus serviços.

O cliente será identificado, pela introdução do seu número de telemóvel no terminal do posto de abastecimento, número que deve conter o indicativo do país, e pelo PIN previamente gerado na app móvel LUSOPAY.

Estes dados deverão ser processados pelo broker e enviados para os serviços da LUSOPAY. Assim, deverá ser invocado o método **initializePaymentTransaction**.

2.3. Serviço MBWay

O que é a MBWay?

É um serviço novo que a LUSOPAY disponibiliza para os parceiros poderem integrar o sistema da SIBS MBWay com os seus terminais POS ou TPA para que o cliente possa pagar por este método através do seu smartphone.

Como funciona?

O comerciante faz um pedido de pagamento para o número do smartphone da pessoa que vai receber esse pedido, é notificada através da aplicação da SIBS MBWay, essa pessoa tem duas opções pagar ou rejeitar. Caso aceite o pedido o comerciante vai receber uma notificação que o pedido foi aceite e pago, caso contrário receber a dizer que foi cancelado.

Outro cenário é o cancelamento desse pedido, caso o pedido não for aceite em 5 minutos é automaticamente cancelado e notificado o cliente que recebeu esse pedido.

O pedido vai ser único caso o pedido seja cancelado é impossível recuperá-lo tem de fazer novamente outro pedido.

As imagens seguintes mostram dois exemplos de possíveis terminais que se pode integrar.



Este método pode ser integrado em aplicações móveis, loja online, POS (etc).

Como funciona internamente?

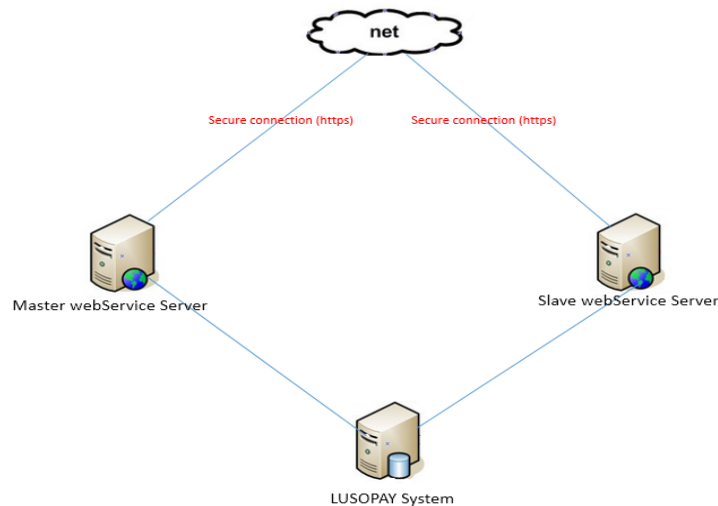
Quando fizer um pedido invocando o método `sendMBWayRequest` passando os devidos parâmetros, esse pedido é guardado pela LUSOPAY numa tabela para depois comunicar e ter registo do pedido caso for aceite ou não. Temos um serviço que vai estar à escuta caso para passar os 5 minutos e não houver nenhum pagamento e que faz uma atualização à tabela desse cancelamento do pedido.

Em ambiente de testes tem de comunicar à SIBS o IP do terminal, IP da rede Wi-Fi e o número do smartphone que vai utilizar.

Nota: O cliente só poderá receber o pedido e pagar se estiver registado no MBWay.

3. Arquitetura

Todos os webservices da LUSOPAY, dada a sensibilidade dos dados que transacionam, operam sobre protocolo seguro (https), e disponibilizam um sistema redundante de tratamento de falhas/indisponibilidade.



Img. 1 – Arquitetura webService

Desta forma, aconselhamos que no consumo do webService a ser descrito, seja implementada a verificação de disponibilidade do servidor “Master” e no caso de indisponibilidade ou demora no seu contacto deverão redirecionar o pedido para o servidor “Slave”.

Os endereços dos webservices “Master” e “Slave” são, respetivamente:

- SOAP 1.1
 - Master Url: <https://services.lusopay.com/PaymentServices/PaymentServices.svc>
 - Slave Url : <https://services.lusopay.com/PaymentServices2/PaymentServices.svc>

Adicionalmente a LUSOPAY disponibiliza aos seus clientes um ambiente de testes, para que o possam utilizar enquanto desenvolvem e configuram os seus sistemas. O endereço para o ambiente de testes é:

- SOAP 1.1
 - Url: https://services.lusopay.com/PaymentServices_test/PaymentServices.svc

Nesse sentido, para utilização do ambiente de teste os clientes deverão informar a LUSOPAY dessa necessidade, de maneira a que se configure devidamente o ambiente e sejam também agendados testes de aceitação para uma posterior passagem ao ambiente produtivo.

4. Métodos wallet

Ao longo dos próximos subcapítulos serão apresentados os métodos que integram o webService.

A autenticação e validação do cliente será realizada com recurso a conjuntos de dois parâmetros, nomeadamente, *key* e *vatNumber*, para métodos que operem sobre pacotes de referências ou, *clientGuid* e *vatNumber*, para métodos que operem sobre referências dinâmicas.

Assim, e dado que estes parâmetros são usados em diversos métodos, será apresentado de forma geral o seu significado.

- clientGuid – Chave para identificação do cliente. Fornecida pela LUSOPAY.
- vatNumber – Num. de contribuinte do cliente para validar se o pacote registado e identificado pelo parâmetro *key/clientGuid*, foi registado para um cliente com o num. de contribuinte fornecido neste parâmetro.

4.1. sendPaymentRequest

Descrição	Método que opera sobre a rede de pagamentos LUSOPAY, permitindo o envio de pedidos de pagamento entre os seus utilizadores;
Assinatura	<code>Function sendPaymentRequest(ByVal clientGuid As String, ByVal vatNumber As String, ByVal toUserName As String, ByVal amount As Double, ByVal description As String, ByVal expirationDate As DateTime, ByVal pos As String) As SendPaymentRequestFeedback</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
toUserName	String	
amount	Double	
description	String	
expirationDate	DateTime	
pos	String	

Retorno

O método devolve um objeto do tipo *SendPayment*, o qual é composto por dois atributos do tipo *String*, nomeadamente *code* e *transactionNumber*.

O atributo *code* deve ser analisado de acordo com a seguinte tabela.

Apenas no caso do atributo *code* ter o valor "OK", deve ser considerado o atributo *transactionNumber*.

Code	Descrição
OK	Operação correu com sucesso
FROM_USER_NOT_FOUND	O nome de utilizador enviado para efetuar o pedido, não existe na rede LUSOPAY
TO_USER_NOT_FOUND	O nome de utilizador enviado para receber o pedido, não existe na rede LUSOPAY
UNABLE_TO_PROCESS_REQUEST	Erro interno, devendo neste caso de utilizar o ambiente redundante
UNABLE_TO_REGISTER_NOTIFICATION_REQUEST	O pedido de pagamento foi efetuado com sucesso, mas o registo do pedido de notificação de pagamento devolveu erro
EXPIRATION_DATE_MUST_BE_IN_FUTURE	A data de expiração do pedido de pagamento é inferior à data do servidor
MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid e vatNumber não correspondem a uma configuração com permissões de utilização do serviço

4.2. cancelPaymentRequest

Descrição	Método criado para cancelar um pedido de pagamento;
Assinatura	<code>Function cancelPaymentRequest(ByVal clientGuid As String, ByVal vatNumber As String, ByVal ownerUserName As String, ByVal transactionNumber As String, ByVal comments As String) As String</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
ownerUserName	String	
transactionNumber	String	
Comments	String	

Retorno

O método devolve uma *String*, a qual deve ser analisada de acordo com a seguinte tabela:

Retorno	Descrição
OK	O pedido de pagamento foi cancelado com sucesso
MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid e vatNumber não correspondem a uma configuração com permissões de utilização do serviço
TRANSACTION_NUMBER_NOT_FOUND	O número de transação passado ao método não corresponde a nenhuma transação ou a transação a que corresponde não é um pedido de pagamento
UNABLE_TO_PROCESS_REQUEST	Erro interno, devendo neste caso de utilizar o ambiente redundante

4.3. listPaymentRequests

Descrição	Método criado para devolver uma lista de pedidos de pagamento;
Assinatura	<code>Function listPaymentRequests(ByVal clientGuid As String, ByVal vatNumber As String, ByVal startDate As DateTime, ByVal endDate As DateTime, ByVal status As List(Of Lusopay.PaymentRequestStatus)) As DataSet</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
startDate	DateTime	
endDate	DateTime	
status	List(Of PaymentRequestStatus))	Se enviado nulo ou sem elementos, considera todos os estados possíveis

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
transaction_number	String	Nº da transação associada ao pedido de pagamento
name	String	Nome do recetor do pedido de pagamento
username	String	Username do recetor do pedido de pagamento
amount	Decimal	Valor do pedido de pagamento
date	DateTime	Data em que o pedido de pagamento foi efetuado
description	String	Descrição introduzida no pedido de pagamento
expiration_date	DateTime	Data de expiração do pedido
status	String	Estado do pedido, de acordo com a enumeração <i>PaymentRequestStatus</i>
pos	String	Código do POS do cliente no caso de este ter sido enviado
fees	Decimal	Valor das comissões associado ao pagamento do pedido

4.4. paymentRequestDetails

Descrição	Método criado para devolver o detalhe de um dado pedido de pagamento;
Assinatura	<code>Function paymentRequestDetails(ByVal clientGuid As String, ByVal vatNumber As String, ByVal transactionNumber As String) As DataSet</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
transactionNumber	String	

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
name	String	Nome do recetor do pedido de pagamento
username	String	Username do recetor do pedido de pagamento
amount	Decimal	Valor do pedido de pagamento
date	DateTime	Data em que o pedido de pagamento foi efetuado
description	String	Descrição introduzida no pedido de pagamento
expiration_date	DateTime	Data de expiração do pedido
status	String	Estado do pedido, de acordo com a enumeração PaymentRequestStatus
pos	String	Código do POS do cliente no caso de este ter sido enviado
fees	Decimal	Valor das comissões associado ao pagamento do pedido

4.5. checkUserVatNumber

Descrição	Método para validar se um dado número de contribuinte está associado a um dado utilizador;
Assinatura	<code>Function checkUserVatNumber(ByVal clientGuid As String, ByVal vatNumber As String, ByVal userName As String, ByVal userVatNumber As String) As String</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
userName	String	Username do utilizador a validar
userVatNumber	String	Número de contribuinte do utilizador a validar

Retorno

O método devolve uma *String*, a qual deve ser analisada de acordo com a seguinte tabela:

Retorno	Descrição
---------	-----------

OK	O número de contribuinte corresponde ao utilizador
NOK	O número de contribuinte não corresponde ao utilizador
MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid e vatNumber não correspondem a uma configuração com permissões de utilização do serviço

4.6. getUserStatus

Descrição	Método criado para validar se um dado username existe na rede LUSOPAY e qual o seu estado;
Assinatura	<code>getUserStatus(ByVal clientGuid As String, ByVal vatNumber As String, ByVal userName As String) As String</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
userName	String	

Retorno

O método devolve uma *String*, a qual deve ser analisada de acordo com a seguinte tabela:

Retorno	Descrição
ACTIVE	O username foi encontrado e é um utilizador com capacidade de efetuar transações na rede LUSOPAY
BLOCKED	O username foi encontrado, no entanto a conta do utilizador encontra-se temporariamente bloqueada, possivelmente devido a erro na introdução das credenciais, mais de três vezes seguidas
DISABLED	O username foi encontrado, no entanto a conta do utilizador encontra-se desabilitada, não podendo por isso, o utilizador entrar na rede LUSOPAY. Este estado pode ser reversível
REMOVED	O username foi encontrado, no entanto a conta do utilizador encontra-se desabilitada, não podendo por isso, o utilizador entrar na rede LUSOPAY. Este estado é irreversível
USER_NOT_FOUND	O username não existe na rede LUSOPAY
MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid e vatNumber não correspondem a uma configuração com permissões de utilização do serviço

4.7. addAwaitingPayment

Descrição	Método que adiciona um pedido de pagamento pendente por parte de um comerciante;
Assinatura	addAwaitingPayment(ByVal clientGuid As String, ByVal vatNumber As String, ByVal pos As String, ByVal amount As Double, ByVal description As String, ByVal externalReference As String) As String

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
pos	String	Código do POS, que identifica o posto de venda.
amount	Double	
description	String	
externalReference	String	Código interno do comerciante que identifica a sua transação. Pode ser enviado vazio, caso não tenha aplicabilidade.

Retorno

O método devolve uma *String*, a qual deve ser analisada de acordo com a seguinte tabela:

Retorno	Descrição
OK	O pedido de pagamento foi registado com sucesso
MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid e vatNumber não correspondem a uma configuração com permissões de utilização do serviço
PENDING_PAYMENT	Já existe um pagamento em espera para o POS especificado
NOK	Erro interno, devendo neste caso de utilizar o ambiente redundante

4.8. getAwaitingPayment

Descrição	Método que devolve os dados do último pedido de pagamento pendente enviado por parte de um comerciante, num dado POS;
Assinatura	getAwaitingPayment(ByVal username As String, ByVal pos As String) As DataSet

Parâmetros de entrada

Nome	Tipo de dados	Descrição
username	String	Username do comerciante.
pos	String	Código do POS, que identifica o posto de venda.

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
amount	Decimal	Valor do pagamento, enviado pelo comerciante.
description	String	

4.9. cancelAwaitingPayment

Descrição	Método que permite cancelar um pedido de pagamento pendente por parte de um comerciante;
Assinatura	<code>cancelAwaitingPayment(ByVal clientGuid As String, ByVal vatNumber As String, ByVal pos As String) As String</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
pos	String	Código do POS, que identifica o posto de venda.

Retorno

O método devolve uma *String*, a qual deve ser analisada de acordo com a seguinte tabela:

Retorno	Descrição
OK	O pedido de pagamento foi cancelado com sucesso
MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid e vatNumber não correspondem a uma configuração com permissões de utilização do serviço
NOK	Erro interno, devendo neste caso de utilizar o ambiente redundante

4.10. commitAwaitingPayment

Descrição	Método que permite dar como concluído um pedido pendente de pagamento;
Assinatura	<code>commitAwaitingPayment(ByVal clientGuid As String, ByVal vatNumber As String, ByVal username As String, ByVal pos As String, ByVal transactionNumber As String) As String</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
username	String	Username do comerciante.
pos	String	Código do POS, que identifica o posto de venda.
transactionNumber	String	Número que identifica a transação de pagamento Lusopay;

Retorno

O método devolve uma *String*, a qual deve ser analisada de acordo com a seguinte tabela:

Retorno	Descrição
OK	O pedido de pagamento foi concluído com sucesso
MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid e vatNumber não correspondem a uma configuração com permissões de utilização do serviço
NOK	Erro interno, devendo neste caso de utilizar o ambiente redundante

4.11. getAwaitingPaymentStatus

Descrição	Método que devolve o estado de um dado pagamento num POS;
Assinatura	getAwaitingPaymentStatus(ByVal clientGuid As String, ByVal vatNumber As String, ByVal pos As String, ByVal externalReference As String) As DataSet

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
pos	String	Código do POS, que identifica o posto de venda.
externalReference	String	Código interno que identifica a transação no comerciante. Pode ser enviado vazio, sendo que nesse caso irá devolver o estado do último pedido de pagamento no POS.

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a estrutura apresentada de seguida. De notar que este método irá devolver sempre um única linha correspondente ao estado do pedido de pagamento mais recente de uma dada *externalReference*, caso a mesma seja enviada, ou do último pedido de pagamento enviado.

Nome da coluna	Tipo de dados	Descrição
pos	String	
amount	Decimal	
description	String	
externalReference	String	
transactionNumber	String	
status	String	WAITING, CANCELED, PAYED;
date	DateTime	Data/Hora em que o pedido obteve o estado;

4.12. preparePaymentTransaction

Descrição	Método para gerar um pin de transação a ser utilizado em sistemas de pagamento com reserva de valor, como por exemplo, os postos de abastecimento.
------------------	--

Assinatura	<code>preparePaymentTransaction(ByVal userCredentials As String) As PreparePaymentTransactionFeedback</code>
-------------------	--

Parâmetros de entrada

Nome	Tipo de dados	Descrição
userCredentials	String	Credenciais para autenticação do utilizador. Este parâmetro de conter o username e a password separados por (:) e codificados em base 64. Ex: <code>Convert.ToBase64String(Encoding.ASCII.GetBytes("username:password"))</code>

Retorno

O método devolve um objeto do tipo *PreparePaymentTransactionFeedback*, o qual é composto por dois atributos do tipo *String*, nomeadamente *code* e *transactionPin*. O atributo *code* deve ser analisado de acordo com a seguinte tabela. Apenas no caso do atributo *code* ter o valor "OK", deve ser considerado o atributo *transactionPin*.

Code	Descrição
OK	
USER_HAS_PENDINGS_TRANSACTIONS	
UNEXPECTED_ERROR	

4.13. initializePaymentTransaction

Descrição	Método para iniciar uma transação com reserva de valor, como por exemplo, os postos de abastecimento. A partir deste momento a conta do utilizador ficar num estado que não pode ser transacionada até o comerciante der indicação da conclusão da transação.
Assinatura	<code>initializePaymentTransaction(ByVal clientGuid As String, ByVal vatNumber As String, ByVal cellPhoneNumber As String, ByVal transactionPin As String) As InitializePaymentTransactionFeedback</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	Será a partir deste parâmetro que se irá identificar o beneficiário da transação, bem como verificar se tem autorização para consumir este método.
vatNumber	String	Num. de contribuinte do broker. Será utilizado para validar se o clientGuid foi emitido para este num. de contribuinte.

cellPhoneNumber	String	O núm. de telemóvel que o cliente inseriu no terminal do posto de abastecimento. Deve incluir o indicativo do país.
transactionPin	String	Código numérico de 4 dígitos que irá autenticar a operação. Este código foi previamente gerado pelo cliente na app LUSOPAY.

Retorno

O método devolve um objeto do tipo *InitializePaymentTransactionFeedback*, o qual é composto por dois atributos, um do tipo *String*, *code*, e um do tipo *Decimal*, *limitAmount*. O atributo *code* deve ser analisado de acordo com a seguinte tabela. Apenas no caso do atributo *code* ter o valor "OK", deve ser considerado o atributo *limitAmount*.

Code	Descrição
CELL_PHONE_NUMBER_NOT_FOUND	O número de telemóvel introduzido pelo cliente no terminal não foi encontrado em nenhuma das contas LUSOPAY.
USER_HAS_PENDINGS_TRANSACTIONS	O cliente não pode executar esta ação porque a transação já se encontra inicializada.
AUTHENTICATION_FAILED	O conjunto número de telemóvel, Pin de transação não correspondem aos valores gerados no passo anterior. Pode acontecer que o período entre a geração do Pin de transação e o início da transação seja superior a 10 min, sendo nesse caso apresentado este erro, devendo ser gerado um novo Pin.
MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid não é válido.
UNEXPECTED_ERROR	Um erro não esperado ocorreu, devendo o broker tentar fazer o re-try da operação no endereço de redundância disponibilizado para o efeito.
INSUFFICIENT_BALANCE	O cliente não dispõe de fundos na sua conta LUSOPAY.
OK	Operação efetuada com sucesso, sendo devolvido o valor limite para abastecimento.

4.14. commitPaymentTransaction

Descrição	Método para confirmar uma transação com reserva de valor, como por exemplo, os postos de abastecimento. Com a invocação deste método será retirado o valor da compra do saldo do cliente e a sua conta será desbloqueada.
Assinatura	commitPaymentTransaction(ByVal clientGuid As String, ByVal vatNumber As String, ByVal cellPhoneNumber As String, ByVal transactionPin As String, ByVal amount As Decimal) As String

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	Será a partir deste parâmetro que se irá identificar o beneficiário da transação, bem como verificar se tem autorização para consumir este método.
vatNumber	String	Num. de contribuinte do broker. Será utilizado para validar se o clientGuid foi emitido para este num. de contribuinte.
cellPhoneNumber	String	O núm. de telemóvel que o cliente inseriu no terminal do posto de abastecimento. Deve incluir o indicativo do país.
transactionPin	String	Código numérico de 4 dígitos que irá autenticar a operação. Este código foi previamente gerado pelo cliente na app LUSOPAY.

Retorno

O método devolve uma *String* representando o estado do processamento do método, de acordo com a seguinte tabela.

Code	Descrição
CELL_PHONE_NUMBER_NOT_FOUND	O número de telemóvel introduzido pelo cliente no terminal não foi encontrado em nenhuma das contas LUSOPAY.
USER_HAS_PENDINGS_TRANSACTIONS	O cliente não pode executar esta ação porque a transação já se encontra inicializada.
AUTHENTICATION_FAILED	O conjunto número de telemóvel, Pin de transação não correspondem aos valores gerados no passo anterior. Pode acontecer que o período entre a geração do Pin de transação e o início da transação seja superior a 10 min, sendo nesse caso apresentado este erro, devendo ser gerado um novo Pin.

MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid não é válido.
UNEXPECTED_ERROR	Um erro não esperado ocorreu, devendo o broker tentar fazer o re-try da operação no endereço de redundância disponibilizado para o efeito.
OK	Operação efetuada com sucesso, sendo devolvido o valor limite para abastecimento.

LUSOPAY

4.15. cancelPaymentTransaction

Descrição	Método para cancelar uma transação com reserva de valor, como por exemplo, os postos de abastecimento. A partir deste momento a conta do utilizador volta a ficar desbloqueada.
Assinatura	cancelPaymentTransaction(ByVal clientGuid <i>As String</i> , ByVal vatNumber <i>As String</i> , ByVal cellPhoneNumber <i>As String</i> , ByVal transactionPin <i>As String</i>) <i>As String</i>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	Será a partir deste parâmetro que se irá identificar o beneficiário da transação, bem como verificar se tem autorização para consumir este método.
vatNumber	String	Num. de contribuinte do broker. Será utilizado para validar se o clientGuid foi emitido para este num. de contribuinte.
cellPhoneNumber	String	O núm. de telemóvel que o cliente inseriu no terminal do posto de abastecimento. Deve incluir o indicativo do país.
transactionPin	String	Código numérico de 4 dígitos que irá autenticar a operação. Este código foi previamente gerado pelo cliente na app LUSOPAY.

Retorno

O método devolve uma *String* representando o estado do processamento do método, de acordo com a seguinte tabela.

Code	Descrição
CELL_PHONE_NUMBER_NOT_FOUND	O número de telemóvel introduzido pelo cliente no terminal não foi encontrado em nenhuma das contas LUSOPAY.
USER_HAS_PENDINGS_TRANSACTIONS	O cliente não pode executar esta ação porque a transação já se encontra inicializada.
AUTHENTICATION_FAILED	O conjunto número de telemóvel, Pin de transação não correspondem aos valores gerados no passo anterior. Pode acontecer que o período entre a geração do Pin de transação e o início da transação seja superior a 10 min, sendo nesse caso apresentado este erro, devendo ser gerado um novo Pin.
MISSING_EXECUTION_PERMISSIONS	O parâmetro clientGuid não é válido.

UNEXPECTED_ERROR	Um erro não esperado ocorreu, devendo o broker tentar fazer o re-try da operação no endereço de redundância disponibilizado para o efeito.
OK	Operação efetuada com sucesso, sendo devolvido o valor limite para abastecimento.

5. Métodos serviços de referências

Ao longo dos próximos subcapítulos serão apresentados os métodos que integram o webService no que diz respeito à sua interação com referências Multibanco e Payshop. A autenticação e validação do cliente será realizada com recurso a conjuntos de dois parâmetros, nomeadamente, *key* e *vatNumber*, para métodos que operem sobre pacotes de referências ou, *clientGuid* e *vatNumber*, para métodos que operem sobre referências dinâmicas.

Assim, e dado que estes parâmetros são usados em diversos métodos, será apresentado de forma geral o seu significado.

- Key – Chave para identificação do pacote de referências e para autenticação no webService. Fornecida pela LUSOPAY.
- clientGuid – Chave para identificação do cliente. Fornecida pela LUSOPAY.
- vatNumber – Num. de contribuinte do cliente para validar se o pacote registado e identificado pelo parâmetro *key/clientGuid*, foi registado para um cliente com o num. de contribuinte fornecido neste parâmetro.

5.1. getMovementsByUser

Descrição	Método criado para devolver os movimentos transacionados na conta LUSOPAY, do cliente identificado pelos parâmetros <i>clientGuid</i> e <i>vatNumber</i> . Este método opera sobre o sistema de contas de pagamento, retornando tanto dados referente a pagamentos de referências como a outras transações efetuadas com suporte da rede LUSOPAY;
Assinatura	<code>getMovementsByUser(ByVal clientGuid As String, ByVal vatNumber As String, ByVal movementType As LUSOPAY.MovementTypes, ByVal numOfItemsPerPage As Integer, ByVal pageNumber As Integer, ByVal startDate As DateTime, ByVal endDate As DateTime) As DataSet</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
movementType	MovementTypes	
numOfItemsPerPage	Integer	
pageNumber	Integer	
startDate	DateTime	
endDate	DateTime	

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
date	DateTime	
amount	Decimal	
description	String	
paymentUsername	String	

LUSOPAY

5.2. getNewReference

Descrição	Método utilizado para devolver uma referência dentro do pacote de referências contratado pelo cliente;
Assinatura	<code>getNewReference(ByVal key As String, ByVal vatNumber As String, ByVal amount As Double, ByVal referenceId As String, ByVal sendEmail As Boolean) As String</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
key	String	
vatNumber	String	
amount	Double	
referenceId	String	Id que irá ficar associado à descrição da referência gerada. Apenas poderá ser interpretado pelo cliente, uma vez que irá representar um objeto no sistema que invoca o serviço.
sendEmail	Boolean	Em caso afirmativo, notifica por email o cliente, indicando da geração de uma referência.

Retorno

O método devolve um objeto do tipo *String*, representando a referência gerada no âmbito do pacote associado.

5.3. getNewMovements

Descrição	Método desenvolvido para permitir obter a lista de movimentos (referências pagas), que ainda não foram integradas nos sistemas do cliente que está a invocar este método. Este método apenas pesquisa nas referências geradas no âmbito do pacote identificado pelos parâmetros <i>key</i> e <i>vatNumber</i> . Devolve todas as referências pagas que tenham um id superior ao parâmetro <i>lastId</i> ;
Assinatura	<code>getNewMovements(ByVal key As String, ByVal vatNumber As String, ByVal lastId As Long) As DataSet</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
key	String	
vatNumber	String	
lastId	Long	

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
clientPaymentDateTime	DateTime	
transactionCost	Decimal	
transactionDateTime	DateTime	
reference	String	
value	Decimal	

5.4. getPaymentDate

Descrição	Método desenvolvido para devolver a data em que a LUSOPAY, efetuou o retorno dos fundos para o cliente;
Assinatura	getPaymentDate(ByVal key As String, ByVal vatNumber As String, ByVal id As String) As String

Parâmetros de entrada

Nome	Tipo de dados	Descrição
key	String	
vatNumber	String	
id	Long	

Retorno

O método devolve um objeto do tipo *String*, representando a data em que a LUSOPAY enviou os fundos relativos ao pagamento da referência para o cliente.

5.5. getKey

Descrição	Método desenvolvido, para permitir recuperar a chave de autenticação nos pacotes de referências. Para esse efeito será necessário conhecer o intervalo de referências contratado;
Assinatura	getKey(ByVal vatNumber <i>As String</i> , ByVal lowerLimit <i>As String</i> , ByVal upperLimit <i>As String</i>) <i>As String</i>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
vatNumber	String	
lowerLimit	String	
upperLimit	String	

Retorno

O método devolve um objeto do tipo *String*, representando a chave associada ao pacote de referências.

5.6. getNewDynamicReference

Descrição	Método para criar uma nova referência dinâmica;
Assinatura	<code>getNewDynamicReference(ByVal clientGuid As String, ByVal vatNumber As String, ByVal valueList As List(Of References), ByVal sendEmail As Boolean) As List(Of ReferencesGenerationFeedback)</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
valueList	List(Of References)	
sendEmail	Boolean	

Retorno

O método devolve um objeto do tipo *List(Of ReferencesGenerationFeedback)*, representando a lista de referências geradas.

Para consultar o resultado da geração das referências deve observar o atributo referenceMB, referencePS e message, de cada um dos itens da lista de retorno.

Caso o valor dos atributos *referenceMB* ou *referencePS* seja igual a -1, significa que a geração da referência não foi processada devidamente, devendo neste caso analisar o atributo *message*, para proceder ao despiste da causa para essa falha.

Assim, as mensagens programadas serão devolvidas de acordo com a seguinte tabela.

Mensagem	Descrição
Value is equal or less than 1.2 euros	A utilização do serviço, obriga a que o valor da referência gerada seja superior a 1.2 euros.
Client does not have permissions to this service	O cliente não tem o serviço ativo
Max elements in valueList is 25	Cada pedido de geração de referências está limitado a 25, ou seja, o parâmetro <i>valueList</i> só pode conter no máximo 25 posições
GUID does not match any client	Ocorre quando a GUID passada no webService não encontra correspondência na base de dados LUSOPAY
Internal service problem	Erro interno do serviço. Neste caso deve ser redirecionado o pedido para o ambiente redundante

5.7. getNewDynamicMovements

Descrição	Método desenvolvido para permitir obter a lista de movimentos (referências pagas), que ainda não foram integradas nos sistemas do cliente que está a invocar este método. Este método apenas pesquisa nas referências geradas no âmbito das referências dinâmicas, excluindo portanto, os pacotes associados a clientes específicos. Devolve todas as referências pagas que tenham um id superior ao parâmetro <i>lastId</i> ;
Assinatura	<code>getNewDynamicMovements(ByVal clientGuid As String, ByVal vatNumber As String, ByVal lastId As Long) As DataSet</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
lastId	Long	

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
id	Long	
transactionDateTime	DateTime	Data/Hora que a referência foi paga
clientPaymentDateTime	DateTime	Data/Hora que a LUSOPAY enviou os fundos para o cliente
transactionCost	Double	Comissão cobrada pela LUSOPAY
reference	String	
amount	Double	
serviceId	Integer	Id do serviço associado à referência
description	String	

5.8. getNewDynamicMovementsByDates

Descrição	Método desenvolvido para permitir obter a lista de movimentos (referências pagas), num dado intervalo de datas. Usado para permitir a consolidação dos movimentos, por exemplo, mensalmente. Este método apenas pesquisa nas referências geradas no âmbito das referências dinâmicas, excluindo portanto, os pacotes associados a clientes específicos. Devolve todas as referências pagas que tenham dentro do intervalo de datas passado;
Assinatura	<code>getNewDynamicMovementsByDates(ByVal clientGuid As String, ByVal vatNumber As String, ByVal startDate As DateTime, ByVal endDate As DateTime) As DataSet</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
startDate	DateTime	
endDate	DateTime	

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
id	Long	
transactionDateTime	DateTime	Data/Hora que a referência foi paga
clientPaymentDateTime	DateTime	Data/Hora que a LUSOPAY enviou os fundos para o cliente
transactionCost	Double	Comissão cobrada pela LUSOPAY
reference	String	
amount	Double	
serviceId	Integer	Id do serviço associado à referência
description	String	

5.9. getCheckDigitByReference

Descrição	
Assinatura	<code>getCheckDigitByReference(ByVal key As String, ByVal vatNumber As String, ByVal reference As String, ByVal amount As Double, ByVal referenceId As String, ByVal sendEmail As Boolean) As String</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
key	String	
vatNumber	String	
reference	String	
amount	Double	
referenceId	String	Id que irá ficar associado à descrição da referência gerada. Apenas poderá ser interpretado pelo cliente, uma vez que irá representar um objeto no sistema que invoca o serviço.
sendEmail	Boolean	Em caso afirmativo, notifica por email o cliente, indicando da geração de uma referência.

Retorno

O método devolve um objeto do tipo *String*, representando o checkDigit associado à referência.

5.10. getNewReferencePayshopNet

Descrição	
Assinatura	getNewReferencePayshopNet(<i>ByVal</i> key <i>As String</i> , <i>ByVal</i> vatNumber <i>As String</i> , <i>ByVal</i> amount <i>As Double</i> , <i>ByVal</i> referenceId <i>As String</i> , <i>ByVal</i> limitDate <i>As Date</i> , <i>ByVal</i> sendEmail <i>As Boolean</i>) <i>As String</i>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
key	String	
vatNumber	String	
amount	Double	
referenceId	String	Id que irá ficar associado à descrição da referência gerada. Apenas poderá ser interpretado pelo cliente, uma vez que irá representar um objeto no sistema que invoca o serviço.
limitDate	Date	
sendEmail	Boolean	Em caso afirmativo, notifica por email o cliente, indicando da geração de uma referência.

Retorno

O método devolve um objeto do tipo *String*, representando a referência PayShopNet.

5.11. getBalanceByUser

Descrição	Método criado para devolver os saldos das contas LUSOPAY, do cliente identificado pelos parâmetros <i>clientGuid</i> e <i>vatNumber</i> . Este método opera sobre o sistema de contas de pagamento, retornando o saldos de cada conta que tiver com suporte da rede LUSOPAY;
Assinatura	getBalanceByUser(<i>ByVal</i> clientGuid <i>As String</i> , <i>ByVal</i> vatNumber <i>As String</i>) <i>As DataSet</i>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
Name	String	Nome da Conta
Balance	Decimal	Saldo da conta

LUSOPAY

6. Métodos públicos

6.1. getBankList

Descrição	Método público que permite a obtenção de uma lista de bancos;
Assinatura	getBankList() As DataSet

Retorno

O método devolve um objeto do tipo *DataSet*, contendo uma tabela com a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
name	String	
link	String	

7. Objectos

7.1. MovementType

Enumeration	
Numeric Value	String Representation
0	All
1	Credit
2	Debit

7.2. References

Nome do atributo	Tipo de dados	Descrição
amount	Double	
description	String	
serviceType	ReferenceType	
limitDate	Date	Só aplicável caso o tipo de referência selecionado no atributo <i>serviceType</i> , for PayShop(PS)
Email	String	

7.3. ReferenceType

Enumeration	
Numeric Value	String Representation
0	Both
1	MB
2	PS

7.4. ReferencesGenerationFeedback

Nome do atributo	Tipo de dados	Descrição
description	String	
entityMB	String	
referenceMB	String	
referencePS	String	
message	String	

7.5. SendPaymentRequestFeedback

Nome do atributo	Tipo de dados	Descrição
code	String	
transactionNumber	String	

7.6. PaymentRequestStatus

Enumeration	
Numeric Value	String Representation
0	CANCELED
1	DENIED
2	EXPIRED
3	OPEN
4	PROCESSED
5	SCHEDULED

7.7. PreparePaymentTransactionFeedback

Nome do atributo	Tipo de dados	Descrição
code	String	
transactionPin	String	

7.8. InitializePaymentTransaction

Nome do atributo	Tipo de dados	Descrição
code	String	
limitAmount	Decimal	

8. Serviços MBWay

8.1. sendMBWayRequest

Descrição	Método que envia um pedido de pagamento para a aplicação da SIBS chamada MBWay do cliente identificado pelos parâmetros <i>clientGuid</i> e <i>vatNumber</i> para um determinado numero de telemóvel com o valor. Este método opera sobre o sistema da SIBS, retornando todos os dados necessário para validar e guardar numa tabela.
Assinatura	<code>sendMBWayRequest(ByVal clientGuid As String, ByVal vatNumber As String, ByVal cellPhoneNumber As Integer, ByVal amount As String, ByVal externalReference As String, ByVal sendEmail As Boolean) As MBWayRequest</code>

Parâmetros de entrada

Nome	Tipo de dados	Descrição
clientGuid	String	
vatNumber	String	
cellPhoneNumber	Integer	
amount	String	
externalReference	String	Texto livre

sendEmail	Boolean	
-----------	---------	--

Retorno

O método devolve um objeto do tipo *MBWayRequest*, contendo a seguinte estrutura:

Nome da coluna	Tipo de dados	Descrição
amount	Double	Valor que foi pedido
currencyCode	String	Saldo da conta
merchantOperationID	String	Referência da operação
message	String	Mensagem de retorno
statusCode	String	Código do estado do pedido
statusMessage	String	Mensagem de retorno por parte da SIBS
timeStamp	String	Data e hora da resposta
token	String	Código que identifica o pedido

ANEXOS

ANEXO 1 - Estrutura do QR Code:

A estrutura do QR code é resumida nas condições seguinte e estrutura abaixo apresentada.

A separação dos campos é feita por ; (ponto e vírgula). É proibida a utilização do (:) em todos os campos, excepto na descrição.

Tamanho máximo (caracteres)	Conteúdo do campo	Obrigatório	Observações
1	Cabeçalho		não usado
2	Código de ação	Sim	pode ser 00, 01 ou 02
27	Utilizador (credor)	Sim	
60	id do POS (credor)		
11	Valor		formato: 999999999.99 a separação das casas decimais faz-se por ponto (.)
3	Moeda (código ISO)		Ex: EUR
	Reservado		Proibida a utilização de ponto e vírgula dentro do campo (;)
255	Descrição		

Tabela de códigos de ação

00 Identificação do comerciante

Preenche o Pagamento a utilizador

Esta opção apenas se destina a simplificar ao consumidor a escolha do comerciante a quem vai pagar (em substituição à pesquisa no diretório ou nos favoritos)

Ignora a informação a seguir ao utilizador

Ex: uma caixa pode ter colocado o autocolante com a identificação do comerciante ou o talão de conta pode levar esse QR code apenas com a identificação do comerciante (pequenos comerciantes)

Exemplo de QR Code:

LUSOPAYAgora, o telemóvel é que paga.
www.lusopay.com

Este código QR identifica que o código de ação é «00» e o credor é «joseribeiro».

01 Identificação do comerciante e da transação

Preenche o Pagamento a utilizador

É uma ajuda mais completa ao consumidor, pois ao ler o código automaticamente preenche o nome do comerciante, valor, moeda, descritivo, etc.

Ex: a impressão do talão de despesa leva esse código de barras e assim o utilizador apenas necessita de confirmar.

Exemplo de QR Code:

LUSOPAYAgora, o telemóvel é que paga.
www.lusopay.com

Este QR Code identifica que o código de ação é o «01», o credor é «joseribeiro», o id do POS «1234», valor «10.00», a moeda «EUR», descrição «exemplo de pagamento por leitura de código QR».

02 Procura pedido de pagamento a consumidor não identificado

Procura pedido de pagamento em aberto desse comerciante nesse POS

Para pagamentos no POS no momento da compra (Ex: Restauração de fast food que pretende pagamentos rápidos)

Quando o cliente diz que vai pagar com Lusopay, o POS deve criar um “pedido de pagamento em aberto” a favor desse comerciante/POS, que inclui a identificação do comerciante, do POS, o valor a pagar e um descritivo (que pode ser o nº da fatura)

O cliente lê o QR Code do POS que automaticamente vai procurar o último pedido de pagamento em aberto para esse POS; depois só tem de fazer a confirmação do pagamento, mas antes permite alterar o valor e descrição se quiser e confirma a transação.

ANEXO 2 – Exemplos de programação e manual de callback:

No link abaixo poderá aceder a uma pasta no Dropbox onde poderá consultar e baixar exemplos de programação bem como, o manual de callback.

<https://www.dropbox.com/sh/u9afxsr4rcdvm2z/AAAePkBLix4AbbUJBncu7RnYa?dl=0>