

ANEXO I

easypay –sistema de pagamentos

Especificações técnicas para a
implementação de web services easypay
Get e XML

10 de Abril de 2008
Ver B

Introdução

Para a implementação dos webservices a easypay definiu em alternativa ao SOAP a possibilidade de se efectuar uma integração baseada em Get e XML

Utilização de webservices com base em GET e XML

Um pagamento efectuado através do sistema da easypay tem 3 fases:

- 1 – calculo de uma referência
- 2 – notificação de que uma referência foi paga (a easypay apenas envia um número de documento)
- 3 – pedido de detalhe da referência paga (usando o número de documento fornecido pela easypay)

1- Emissão, Cálculo e Comunicação de uma referência multibanco

O primeiro passo é o cálculo da referência. Uma referência tem 9 dígitos e é composta por 3 partes (CIN+DOC+CheckDigit). CIN identifica de forma inequívoca a EPS, DOC é o número do documento(factura, orçamento ou cliente) e que varia entre 0000 e 9999. Check digit é um valor entre 00 e 99 e que é calculado através do MOD97 tendo como input os 7 primeiros dígitos da referência e o montante .

Pedido de referência disponível

| Nome do campo | Tipo | Descrição |
|----------------|--------|---|
| ep_cin | string | (ex:8889) |
| ep_user | string | Número EPS (ex: EASYTEST9) |
| ep_value | double | valor a pagar. Tem que ser > 1,00 e menor que 99.999,99. Tem que ter 2 casas decimais (ex: 123.99). As casas decimais devem estar separadas por ponto |
| ep_key | string | Chave - Código único e que identifica de forma inequívoca a referência. (ex: serial) |
| ep_type | string | auto – calcula próxima ref disponível doc – calcula ref com base no n. doc fornecido |
| ep_doc | string | Opcional - número de documento a ser calculado na referência. Pode variar entre 000 e 9999. (ex: 001 ou 1) |
| ep_description | string | Opcional –descritivo da referência. Este campo é disponibilizado no backoffice. |
| ep_email | string | Opcional – email para quem esta a ser calculada a referencia. Este campo é disponibilizado no backoffice. Em breve a easypay pode enviar um email com os dados de pagamento. |
| ep_mobile | string | Opcional – número de telefone para quem esta a ser calculada a referência. Este campo é disponibilizado no backoffice. Em breve a easypay pode enviar um sms com os dados de pagamento. |

Resposta

| Nome do campo | Tipo | Descrição |
|---------------|--------|--|
| ep_status | string | ok or err |
| ep_message | string | Informa os testes efectuados e o seu estado. separados por ponto e vírgula |
| ep_cin | string | (ex:8889) |

| | | |
|--------------|--------|--|
| ep_user | string | Número EPS (ex: EASYTEST9) |
| ep_entity | string | 10611 – Nesta fase é fixa mas num futuro próximo será variável |
| ep_reference | string | (ex:888900123 deve ser apresentado no formato xxx xxx xxx) |
| ep_value | double | valor a pagar |
| ep_key | String | Chave - Código único que identifica de forma inequívoca a referência. (ex: serial) |

Exemplo:

https://www.easypay.pt/_s/api_easypay_01AG.php

?ep_cin= 8889

&ep_user= EASYTEST9

&ep_value= 10.00

&ep_doc= 9776

&ep_type= auto

& =teste

[ep_email=teste@easypay.pt](mailto:teste@easypay.pt)

& ep_description =teste

&A09=930000000

A easypay responde com um ficheiro XML que terá o seguinte formato

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<getautoMB >
<ep_status>ok0</ep_status>
<ep_message>id and cin ok - ip ok - automatic reference generated - 888900174 - </ep_message>
<ep_cin>8889</ep_cin>
<ep_user>EASYTEST9</ep_user>
<ep_entity>10611</ep_entity>
<ep_reference>888900174</ep_reference>
<ep_value>10.00</ep_value>
<ep_key>9776</ep_key>
</getautoMB >
```

2- Notificação de pagamento

Sempre que uma referência é paga à easypay é calculado um número de documento que é único. A easypay comunica-o à EPS. Esta comunicação é efectuada através de um GET a uma página específica:

Pedido de chave

| Nome do campo | Tipo | Descrição |
|---------------|--------|--|
| ep_cin | string | (ex:8889) |
| ep_user | string | Número EPS (ex: EASYTEST9) |
| ep_doc | string | Número de documento calculado pela easypay e que identifica de forma inequívoca cada pagamento |

Resposta

| Nome do campo | Tipo | Descrição |
|---------------|--------|--|
| ep_status | string | ok ou err |
| ep_message | string | Informa os testes efectuados e o seu estado. separados por ponto e virgula |
| ep_cin | string | (ex:8889) |
| ep_user | string | Número EPS (ex: EASYTEST9) |

| | | |
|--------|--------|---|
| ep_doc | string | Número de documento calculado pela easypay e que indentifica de forma inequívoca cada pagamento |
| ep_key | String | Chave - Código único que identifica de forma inequívoca a referência. (ex: serial) |

Exemplo:

[http://www.dominio.pt/easypay/api_easypay_02AG.php?](http://www.dominio.pt/easypay/api_easypay_02AG.php?ep_cin= 8889)

?ep_cin= 8889

&ep_user= EASYTEST9

&ep_doc= EASYTEST92008091256378290408

Como resposta a easypay recebe um ficheiro XML que tem que devolver uma chave (que deve ser sequencial).

Como resposta a easypay recebe um ficheiro XML que devolve o detalhe da transacção.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<getautoMB_key>
<ep_status>ok0</ep_status>
<ep_message>doc gerado</ep_message>
<ep_cin>8889</ep_cin>
<ep_user>EASYTEST9</ep_user>
<ep_doc> EASYTEST92008091256378290408</ep_doc>
<ep_key>1</ep_key>
</getautoMB_key>
```

Tabela de suporte

A easypay sugere que exista um tabela que seja comum ao webservice 2G e 3G com os seguintes campos:

```
CREATE TABLE `easypay_autoMB_key` (
  `ep_key` int(11) NOT NULL auto_increment,
  `ep_doc` varchar(50) default NULL,
  `ep_cin` varchar(20) default NULL,
  `ep_user` varchar(20) default NULL,
  `ep_date_stamp` timestamp NULL default CURRENT_TIMESTAMP,
  `ep_status` varchar(20) default 'pending',
  `ep_entity` varchar(10) default NULL,
  `ep_reference` varchar(9) default NULL,
  `ep_value` double default NULL,
  `ep_payment_type` varchar(6) default NULL,
  `ep_value_fixed` double default NULL,
  `ep_value_var` double default NULL,
  `ep_value_tax` double default NULL,
  `ep_value_transf` double default NULL,
  `ep_date_transf` date default NULL,
  `ep_date_read` date default NULL,
  `ep_status_read` varchar(20) default NULL,
  `ep_invoice_number` varchar(30) default NULL,
  `ep_transf_number` varchar(20) default NULL,
  PRIMARY KEY (`ep_key`),
  UNIQUE KEY `ep_doc` (`ep_doc`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

3 – Consulta de detalhe de pagamento

Após a EPS ter sido notificada de que um pagamento foi recebido, pode consultar o detalhe do mesmo. É só aqui que fica a conhecer qual a referência que foi paga, o seu valor, a data da transferência, etc.

Pedido de detalhe de pagamento efectuado

| Nome do campo | Tipo | Descrição |
|---------------|--------|--|
| ep_cin | string | (ex:8889) |
| ep_user | string | Número EPS (ex: EASYTEST9) |
| ep_key O | string | Chave enviada para a easypay como resposta no pedido autoMBkey |
| ep_doc | string | Número de documento enviado pela easypay no pedido autoMBkey |

Resposta

| Nome do campo | Tipo | Descrição |
|-----------------|--------|--|
| ep_status | string | ok or err |
| ep_message | string | Informa os testes efectuados e o seu estado. separados por ponto e vírgula |
| ep_cin | string | (ex:8889) |
| ep_user | string | Número EPS (ex: EASYTEST9) |
| ep_entity | string | 10611 – Nesta fase é fixa mas num futuro próximo será variável |
| ep_reference | string | (ex:888900123 deve ser apresentado no formato xxx xxx xxx) |
| ep_value | double | valor a pagar |
| ep_key | string | Chave enviada para a easypay como resposta no pedido autoMBkey |
| ep_doc | string | Número de documento enviado pela easypay no pedido autoMBkey |
| ep_payment_type | string | MB – Pagamento Referência Multibanco CC – Cartão Crédito DC – Cartão Débito DD – Débito Directo |
| ep_value_fixed | double | valor de serviço fixo |
| ep_value_var | double | valor de serviço variável |
| ep_value_tax | double | valor do IVA |
| ep_value_transf | double | valor líquido a transferir (valor recebido-valor serviços easypay – IVA) |
| ep_date_transf | date | data em que este valor foi ou vai ser transferido |
| ep_date_read | date | data da última leitura deste registo |
| ep_status_read | Date | Verified |

```
https://www.easypay.pt/_s/api_easypay_03AG.php?  
?ep_cin= 8889  
&ep_user= EASYTEST9  
&ep_key=1  
&ep_doc= EASYTEST92008091256378290408
```

Resposta

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<getautoMB_detail>  
<ep_status>ok0</ep_status>
```

```
<ep_message>id e cin ok;ip ok;doc EASYTEST92008091256378290408 and key 1 ok;</ep_message>
<ep_cin>8889</ep_cin>
<ep_user>EASYTEST9</ep_user>
<ep_key>1</ep_key>
<ep_doc> EASYTEST92008091256378290408</ep_doc>
<ep_payment_type>MB</ep_payment_type>
<ep_entity>10611</ep_entity>
<ep_reference>888900174</ep_reference>
<ep_value>10.00</ep_value>
<ep_value_fixed>0.35</ep_value_fixed>
<ep_value_var>0.18</ep_value_var>
<ep_value_tax>0.11</ep_value_tax>
<ep_value_transf>9.36</ep_value_transf>
<ep_date_transf>2008-01-29 00:00:00</ep_date_transf>
<ep_date_read>2008-04-11 20:19:42</ep_date_read>
<ep_status_read>verified</ep_status_read>
</getautoMB_detail>
```

4- Instruções de instalação

- 1 - Criar a tabela easypay_autoMB_key ou outra semelhante;
- 2 - comunicar à easypay o URL absoluto do ficheiro api_easypay_02AG.xxx com a extensão utilizada;
- 3 - comunicar à easypay o IPFixo do servidor que vai ser utilizado;
- 4 - testar as comunicações com os exemplos fornecidos

5- Exemplos

Geração de referência com base num documento

http://www.easypay.pt/_t/api_1G.php

Este exemplo tem como variável o número de documento. Por defeito o número de documento é o 9778.

Geração de referência automática

http://www.easypay.pt/_t/api_G1.php

Notificação de Pagamento

http://www.easypay.pt/_t/api_G2.php

Leitura de detalhe de Pagamento

http://www.easypay.pt/_t/api_G3.php