

## GATEWAY

---

### Guia de Integração Gateway de Pagamentos REDUNIQ

## INDÍCE

---

Introdução	3
Contactos	3
Integração com a aplicação do comerciante	4
Inicializar pagamento web	8
Redirecionar para a página do Gateway de Pagamentos	8
Consultar os dados de um pagamento	8
Ligação ao gateway	9
Requisitos de segurança	9
Ambiente de testes	9
Ambiente de produção	9
Funções da API	10
initPayment	10
getResult	15
doCapture	17
doRefund	19
doVoid	21
ANEXOS	23
Códigos e mensagens	23
Gestão de timeouts	25

Revisões a este documento

<b>Data</b>	<b>Alteração</b>

## Introdução

---

Este documento descreve o procedimento de integração do *Gateway de Pagamentos* da REDUNIQ com a aplicação do comerciante.

A interface *web do Gateway de Pagamentos* permite ao comerciante disponibilizar aos seus clientes um meio para efetuar pagamentos seguros, utilizando a solução de pagamentos Reduniq, PayPal, AMEX, Web2Pay ou SIBS MB WAY.

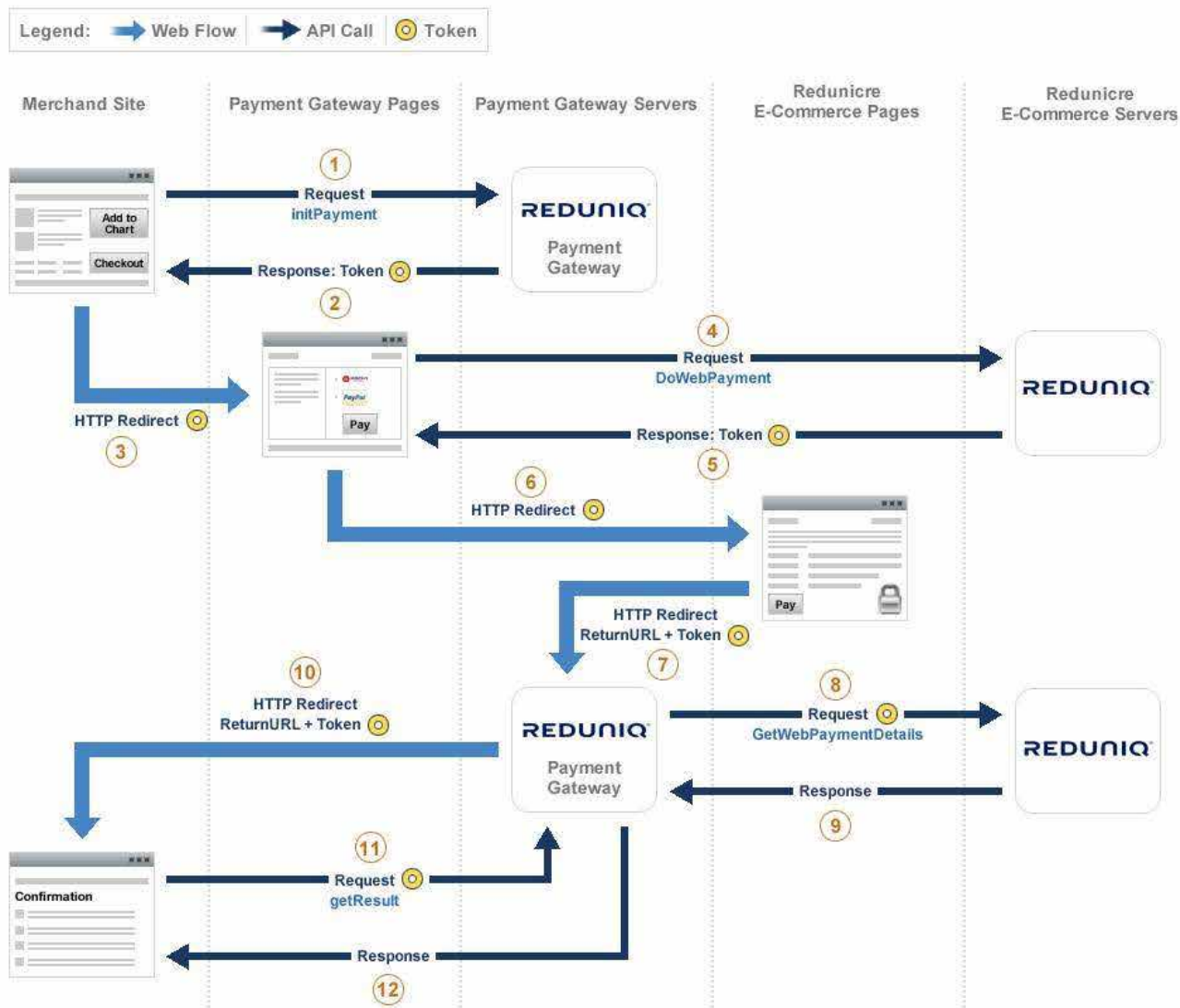
## Contactos

A REDUNIQ disponibiliza os seguintes contactos para dar suporte à integração do *Gateway de Pagamentos*:

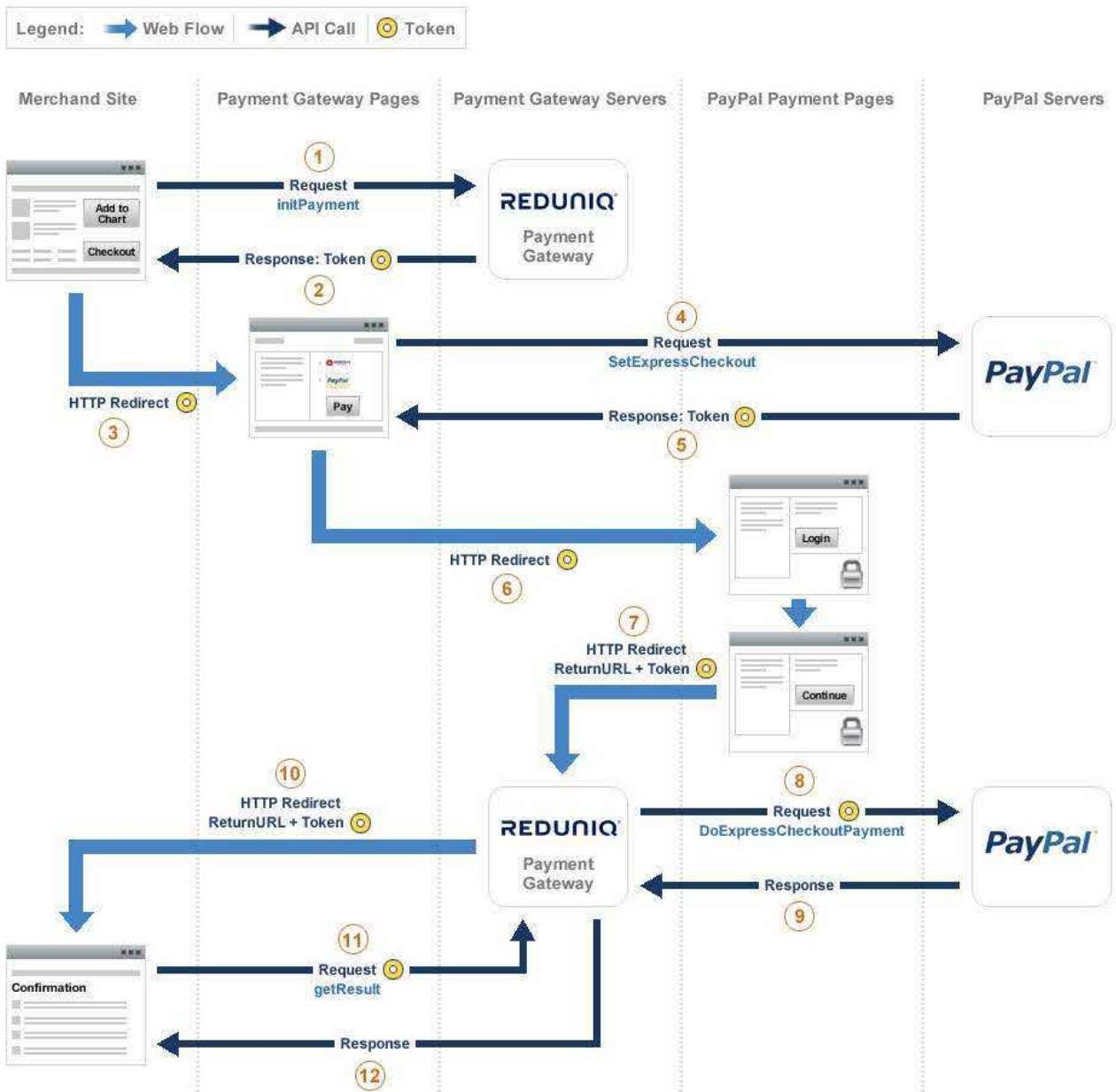
E-mail	Telefone / Fax	Correio
apoio tecnico@unicre.pt	+ 351 21 313 29 00	Av. António Augusto de Aguiar, 122 7º Piso 1050 Lisboa

# Integração com a aplicação do comerciante

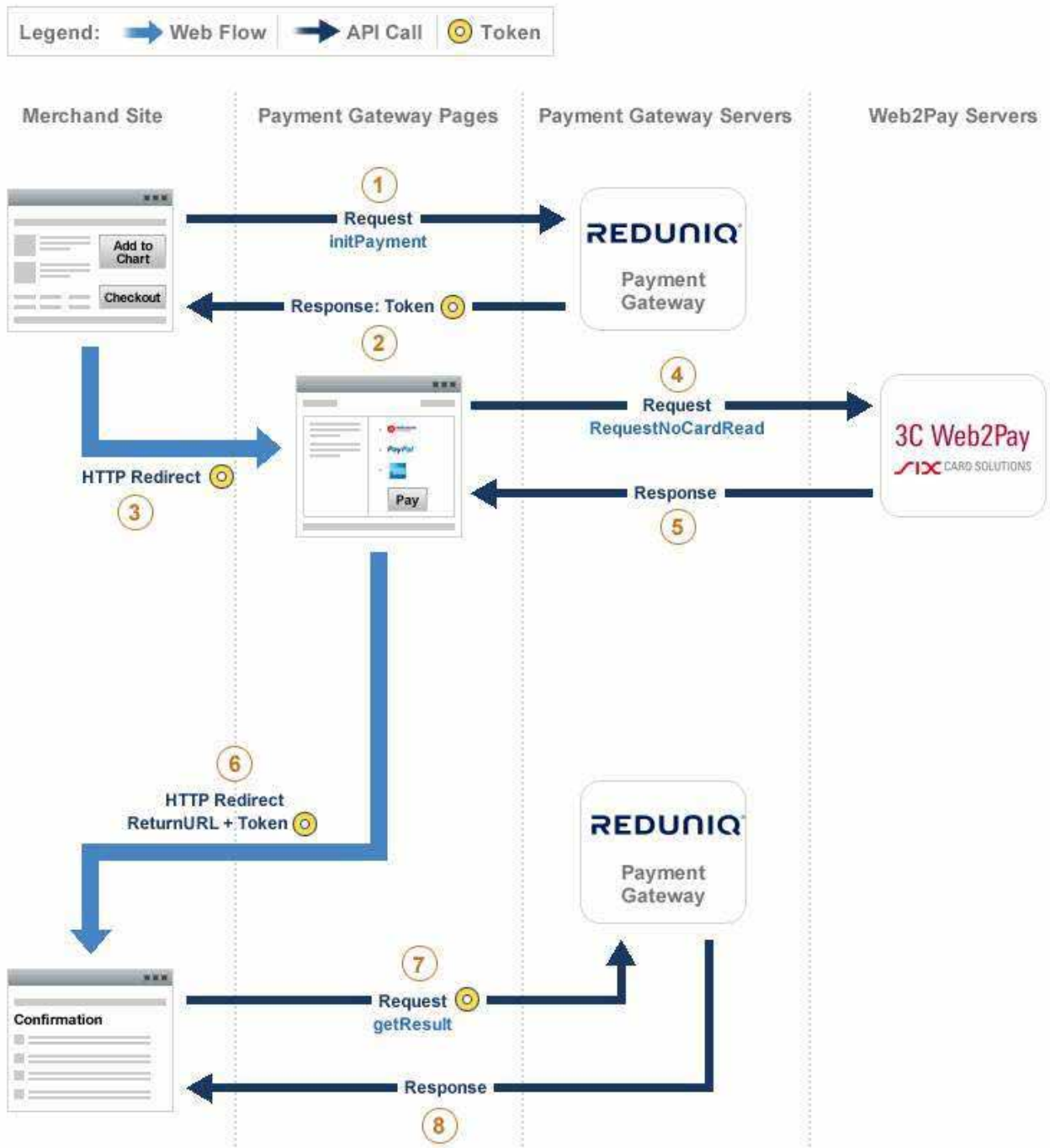
A integração da aplicação de e-commerce com o Gateway de Pagamentos da REDUNIQ disponibiliza, aos clientes do comerciante, modos de pagamentos seguros através do REDUNIQ E-Commerce, PayPal, AMEX Web2Pay ou SIBS MB WAY.



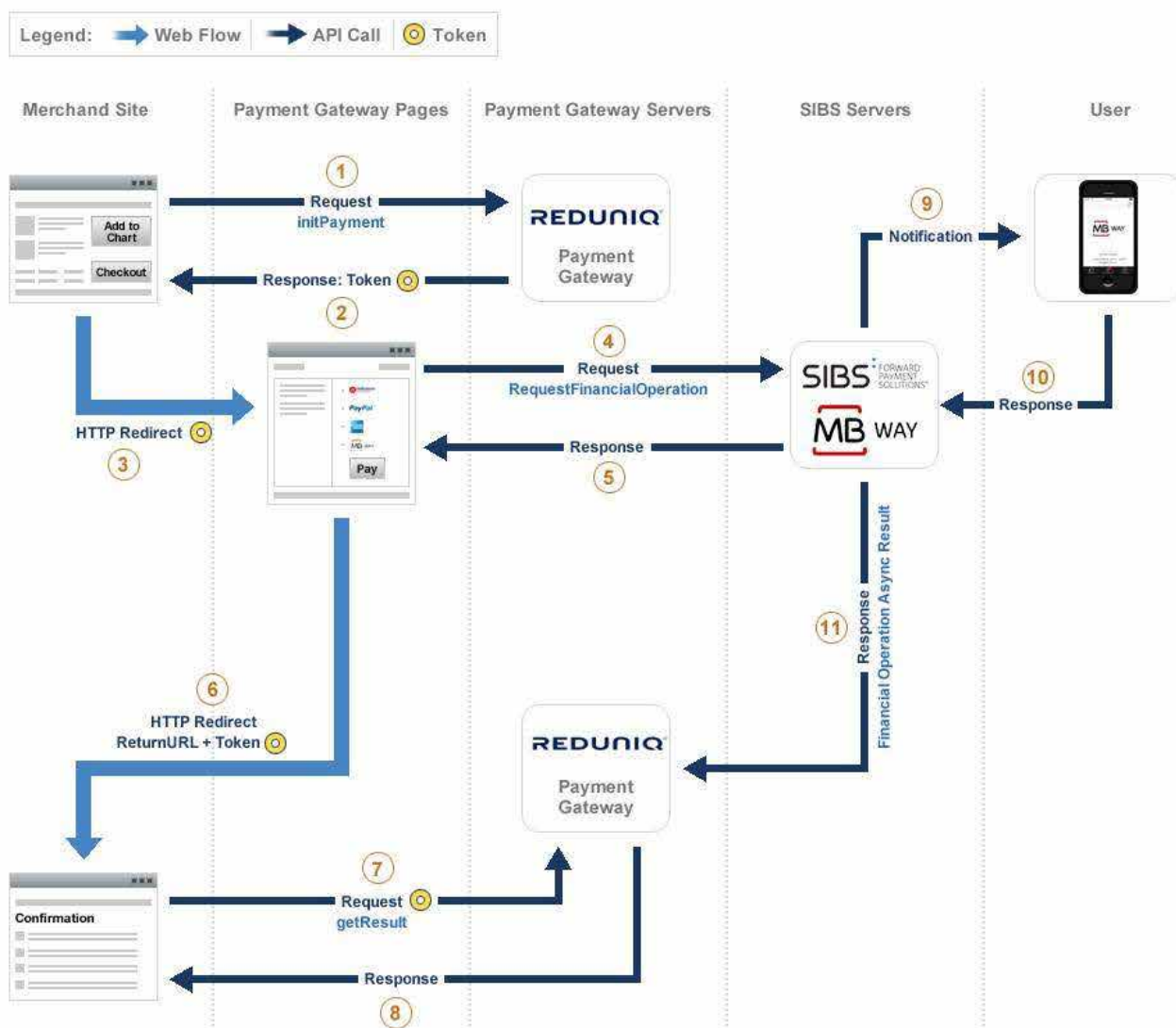
Workflow do processo de pagamento (Reduniq E-Commerce)



Workflow do processo de pagamento (PayPal)



Workflow do processo de pagamento (AMEX Web2Pay)



Workflow do processo de pagamento (SIBS MBWAY)

### Descrição do processo:

1. O site do comerciante chama o *webservice* "initPayment" para iniciar o processo de pagamento. Na chamada ao serviço, para além dos seus dados de identificação (*username e password*), deve indicar os restantes dados necessários para iniciar o processo (nome do cliente, valor a pagar, etc.). A password não é armazenada no *gateway* e serve de chave de descriptação dos dados de acesso às plataformas de pagamentos disponíveis para esse comerciante.
2. O cliente é redirecionado para a página web do gateway onde este seleciona a plataforma de pagamentos a utilizar (REDUNIQ E-Commerce, PayPal, AMEX Web2Pay ou SIBS MB WAY). Caso o comerciante apenas tenha disponível uma plataforma de pagamentos, a página de seleção pode ou não ser exibida ao cliente (para o AMEX Web2Pay e SIBS MB WAY a página é sempre exibida).
3. O *Gateway de Pagamentos* estabelece as comunicações necessárias com a plataforma selecionada.
4. O cliente é redirecionado para a plataforma selecionada onde este efetuará o pagamento indicando os dados do seu cartão VISA/MASTERCARD/AMEX ou utilizando a sua conta PayPal.



5. Após conclusão (ou cancelamento) do pagamento, a plataforma de pagamento utilizada redireciona para o *Gateway de Pagamentos* que registra em log toda a informação associada.
6. A página de retorno do *gateway* chama o *webservice* disponibilizado pela plataforma selecionada para obter a informação associada ao processo e envia os dados relevantes na resposta à aplicação do comerciante.
7. O cliente é redirecionado para a página do comerciante onde será exibida a mensagem de sucesso/erro.
8. No caso do comerciante não receber o retorno num determinado período de tempo, deverá chamar o *webservice* “*getResult*” para verificar o estado do mesmo. Poderá também consultar o backoffice próprio da plataforma de pagamento utilizada.
9. O Utilizador recebe uma notificação na app MB WAY, informando sobre o pedido do *gateway/Comerciante*.
10. O Utilizador pode aceitar ou rejeitar o pedido.
11. O resultado da operação financeira é comunicado ao *gateway* assincronamente.

### **Inicializar pagamento web**

A inicialização de um pagamento web (função *initPayment*), consiste na comunicação ao *Gateway de Pagamentos* de toda a informação necessária para se proceder a um pagamento.

Os campos obrigatórios para iniciar um pagamento *web* estão indicados em “Funções da API”.

### **Redirecionar para a página do Gateway de Pagamentos**

Uma vez iniciado um pagamento, o cliente deve ser redirecionado para a página web do *Gateway de Pagamentos*. Neste processo estão envolvidas as seguintes operações:

- . Na inicialização do pagamento, obter o *token* de sessão e o endereço da página web do *Gateway de Pagamentos*.
- . Associar, no sistema de informação do comerciante, o *token* com a compra.
- . Redirecionar o cliente para a página *web* do *gateway*, através de uma resposta HTTP.

### **Consultar os dados de um pagamento**

Quando o processo de pagamento termina, o *Gateway de Pagamentos* redireciona o cliente para a página do comerciante. A aplicação do comerciante deve chamar sempre a função “*getResult*” para obter os detalhes do pagamento, nomeadamente, o código de resposta.

Com essa informação o comerciante pode informar o seu cliente qual foi o resultado do pagamento (consultar a tabela em anexo “Códigos e mensagens”).

# Ligação ao gateway

---

## Requisitos de segurança

Para garantir que todas as comunicações com o *Gateway de Pagamentos* são seguras deve garantir que:

- . Todas as ligações são feitas com HTTPS/SSL.
- . Nunca divulgue os seus dados de autenticação (API username e API password).
- . Verifique que o certificado do servidor presente na ligação HTTPS pertence à Reduniq e que este é válido (não expirou nem foi revogado).

## Ambiente de testes

Para testar a integração da aplicação do comerciante com o *Gateway de Pagamentos REDUNIQ*, utilize os seguintes endereços:

### **SOAP API:**

Contrato WSDL:

<https://pagamentos.reduniq.pt/sandbox-api/v1.0/soap/?wsdl>

Web Services:

<https://pagamentos.reduniq.pt/sandbox-api/v1.0/soap/>

### **NVP API:**

<https://pagamentos.reduniq.pt/sandbox-api/v1.0/nvp/>

Nota: São aceites variáveis submetidas através do método GET ou POST

Interface web para testar os métodos disponíveis:

<https://pagamentos.reduniq.pt/sandbox-test/>

## Ambiente de produção

Em produção, utilize os endereços:

### **SOAP API:**

Contrato WSDL:

<https://pagamentos.reduniq.pt/api/v1.0/soap/?wsdl>

Web Services:

<https://pagamentos.reduniq.pt/api/v1.0/soap/>

### **NVP API:**

<https://pagamentos.reduniq.pt/api/v1.0/nvp/>

Nota: São aceites variáveis submetidas através do método POST

Interface web para testar os métodos disponíveis:

<https://pagamentos.reduniq.pt/sandbox-test/>

## Funções da API

O *Gateway de Pagamentos REDUNIQ* disponibiliza duas API's, uma que implementa o protocolo SOAP, e outra no formato NVP (*Name/Value Pairs*). Todos os dados enviados à API SOAP devem ser codificados em Unicode (UTF-8). Nas respostas, a API retorna sempre os dados em UTF-8. Os valores da *string* NVP, nos pedidos e respostas da API, estão no formato URL-encoded.

A API disponibiliza as seguintes funções:

Função	Descrição
initPayment	Inicialização de um pagamento web
getResult	Obter os detalhes de um pagamento
doCapture	Aceitar um pagamento autorizado
doRefund	Reembolsar um pagamento autorizado e aceite
doVoid	PayPal: Cancelar um pagamento autorizado SIBS MB WAY: Permite anular uma outra operação financeira submetida

### initPayment

A função *“initPayment”* inicializa um pagamento antes de redirecionar o cliente para a página web do Gateway de Pagamentos, onde este selecionará o meio de pagamento.

#### Estrutura de dados do pedido:

Campo	Descrição	Obrig.	Formato	Observações
api.username	User name da API	sim	AN100	-
api.password	Password da API	sim	AN20	-
payment.amount	Valor total a pagar (order amount + order taxes + shipping amount)	sim	N12	O valor 100 corresponde a 1€ O valor 1599 corresponde a 15,99€
payment.action	Código da ação de pagamento	sim	N3	100 – Autorização 101 – Autorização + Captura
payment.description	Descrição do pagamento	sim	AN255	-
order.ref	Referência do pagamento	sim	AN50	Cada pagamento deve ter uma referência única
order.amount	Valor a pagar (sem impostos e sem transporte)	sim	N12	O valor 100 corresponde a 1€ O valor 1599 corresponde a 15,99€
order.taxes	Valor total dos impostos	não	N12	O valor 100 corresponde a 1€ O valor 1599 corresponde a 15,99€
order.date	Data e hora do pagamento	sim	AN19	2012-12-15 20:30:00
order.shipping	Expedição / entrega	não	N1	0 – Não (por omissão) 1 – Sim Se existe expedição e o order.details não é indicado, é considerado um bem “Físico/Material”

Campo	Descrição	Obrig.	Formato	Observações
order.details	Detalhes do pagamento	não	N1	0 – Não (por omissão) 1 – Sim Se existe expedição e o order.details não é indicado, é considerado um bem “Físico/Material”
buyer.firstName	Nome do cliente	não	AN50	-
buyer.lastName	Sobrenome do cliente	não	AN50	-
buyer.email	E-mail do cliente	sim	AN150	-
buyer.shipping.name	Nome da pessoa associada a este endereço	não	AN32	-
buyer.shipping.street1	Expedição: Rua 1	não	AN100	-
buyer.shipping.street2	Expedição: Rua 2	não	AN100	-
buyer.shipping.city	Expedição: Cidade	não	AN40	-
buyer.shipping.state	Provincia/estado	não	AN40	-
buyer.shipping.zipCode	Expedição: Código postal	não	AN20	-
buyer.shipping.country	Expedição: País	não	AN2	ISO 3166-1 (exemplo: “pt”)
buyer.shipping.phone	Expedição: Telefone	não	AN15	-
buyer.shipping.amount	Expedição: Valor	não	N12	O valor 100 corresponde a 1€ O valor 1599 corresponde a 15,99€ Quando definido, o order.shipping deve ser igual a “1”
returnUrlOk	URL de retorno para a página do comerciante	sim	AN255	Usado quando o pagamento é aceite.
returnUrlError	URL de retorno para a página do comerciante	sim	AN255	Usado quando o pagamento é recusado ou cancelado.
notificationUrl	URL de notificação	não	AN255	-
privateData	Dados privados	não	-	Ver estrutura de dados “privateData”
languageCode	Código do idioma	não	AN6	ISO 639-2 (exemplo: “por”)

### Estrutura de dados “order.details” (0 a 100)

Campo	Descrição	Obrig.	Formato	Observações
name	Nome/código do produto ou serviço	sim	AN50	-
amount	Valor unitário (sem impostos)	não	N12	O valor 100 corresponde a 1€ O valor 1599 corresponde a 15,99€
tax	Valor unitário do imposto	não	N12	O valor 100 corresponde a 1€ O valor 1599 corresponde a 15,99€
quantity	Quantidade	não	N5	-
category	Categoria (Físico/Digital)	não	N1	0 – Digital/Serviço (por omissão) 1 – Físico/Material

## Estrutura de dados "order.details" (0 a 100)

Campo	Descrição	Obrig.	Formato	Observações
name	Nome/chave	sim	N50	Exemplo: "NIF"
valuet	Valor	sim	N255	Exemplo: "508855567"

## Exemplo de um pedido em formato SOAP:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:reduniqPaymentGateway">
  <SOAP-ENV:Body>
    <tns:initPayment xmlns:tns="urn:reduniqPaymentGateway">
      <api xsi:type="tns:apiType">
        <username xsi:type="xsd:string">demouser</username>
        <password xsi:type="xsd:string">abc123</password>
      </api>
      <payment xsi:type="tns:paymentType">
        <amount xsi:type="xsd:integer">12500</amount>
        <action xsi:type="xsd:integer">100</action>
        <description xsi:type="xsd:string">Lorem ipsum dolor sit amet, consectetur</description>
      </payment>
      <order xsi:type="tns:orderType">
        <ref xsi:type="xsd:string">ref1359313622</ref>
        <amount xsi:type="xsd:integer">11500</amount>
        <taxes xsi:type="xsd:integer">500</taxes>
        <date xsi:type="xsd:string">2013-01-27 19:07:02</date>
        <shipping xsi:type="xsd:integer">1</shipping>
        <details xmlns="" xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="tns:orderDetailsItemType[2]">
          <item xsi:type="tns:orderDetailsItemType">
            <name xsi:type="xsd:string">Item #1</name>
            <amount xsi:type="xsd:integer">5500</amount>
            <tax xsi:type="xsd:integer">500</tax>
            <quantity xsi:type="xsd:integer">1</quantity>
            <category xsi:type="xsd:integer">1</category>
          </item>
          <item xsi:type="tns:orderDetailsItemType">
            <name xsi:type="xsd:string">Item #2</name>
            <amount xsi:type="xsd:integer">5500</amount>
            <tax xsi:type="xsd:integer">500</tax>
            <quantity xsi:type="xsd:integer">1</quantity>
            <category xsi:type="xsd:integer">1</category>
          </item>
        </details>
      </order>
      <buyer xsi:type="tns:buyerType">
        <firstName xsi:type="xsd:string">Manuel</firstName>
        <lastName xsi:type="xsd:string">Silva</lastName>
        <email xsi:type="xsd:string">manuel.silva@mail.com</email>
        <shipping xmlns="" xsi:type="tns:shippingType">
          <name xsi:type="xsd:string">Manuel Silva</name>

```

```

    <street1 xsi:type="xsd:string">Rua Alberto Sampaio</street1>
    <street2 xsi:type="xsd:string">n50, 2D</street2>
    <city xsi:type="xsd:string">Lisboa</city>
    <state xsi:type="xsd:string">Lisboa</state>
    <zipCode xsi:type="xsd:string">2100-456</zipCode>
    <country xsi:type="xsd:string">pt</country>
    <phone xsi:type="xsd:string">+351213456784</phone>
    <amount xsi:type="xsd:integer">500</amount>
  </shipping>
</buyer>
<returnUrlOk xsi:type="xsd:string">https://www.merchant.pt/ok/</returnUrlOk>
<returnUrlError xsi:type="xsd:string">https:// www.merchant.pt/error/</returnUrlError>
<notificationUrl xsi:type="xsd:string">https://www.merchant.pt/notification/</notificationUrl>
<privateData xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="tns:privateDataItem Type[3]">
  <item xsi:type="tns:privateDataItem Type">
    <name xsi:type="xsd:string">key #1</name>
    <value xsi:type="xsd:string">value #1</value>
  </item>
  <item xsi:type="tns:privateDataItem Type">
    <name xsi:type="xsd:string">key #2</name>
    <value xsi:type="xsd:string">value #2</value>
  </item>
  <item xsi:type="tns:privateDataItem Type">
    <name xsi:type="xsd:string">key #3</name>
    <value xsi:type="xsd:string">value #3</value>
  </item></privateData>
  <languageCode xsi:type="xsd:string">por</languageCode>
</tns:initPayment>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Exemplo de um pedido no formato NVP:

```

METHOD=initPayment
&API_USERNAME=demouser
&API_PASSWORD=abc123
&PAYMENT_AMOUNT=12500
&PAYMENT_ACTION=100
&PAYMENT_DESCRIPTION=Lorem+ipsum+dolor+sit+amet%2c+consectetur
&ORDER_REF=ref1359313622
&ORDER_AMOUNT=11500
&ORDER_TAXES=500
&ORDER_DATE=2013-01-27+19%3a07%3a02
&ORDER_SHIPPING=1
&ORDER_DETAILS_NAME_0=Item+%231
&ORDER_DETAILS_AMOUNT_0=5500
&ORDER_DETAILS_TAX_0=500
&ORDER_DETAILS_QUANTITY_0=1
&ORDER_DETAILS_CATEGORY_0=1
&ORDER_DETAILS_NAME_1=Item+%232
&ORDER_DETAILS_AMOUNT_1=5500
&ORDER_DETAILS_TAX_1=500
&ORDER_DETAILS_QUANTITY_1=1

```

&ORDER\_DETAILS\_CATEGORY\_1=1  
 &BUYER\_FIRSTNAME=Manuel  
 &BUYER\_LASTNAME=Silva  
**&BUYER\_EMAIL=manuel.silva%40mail.com**  
 &BUYER\_SHIPPING\_NAME=Manuel+Silva  
 &BUYER\_SHIPPING\_STREET1=Rua+Alberto+Sampaio  
 &BUYER\_SHIPPING\_STREET2=n50%2c+2D  
 &BUYER\_SHIPPING\_CITY=Lisboa  
 &BUYER\_SHIPPING\_STATE=Lisboa  
 &BUYER\_SHIPPING\_ZIPCODE=2100-456  
 &BUYER\_SHIPPING\_COUNTRY=pt  
 &BUYER\_SHIPPING\_PHONE=%2b351213456784  
 &BUYER\_SHIPPING\_AMOUNT=500  
**&RETURNURLOK=https%3a%2f%2fwww.merchant.pt%2fok%2f**  
**&RETURNURLERROR=https%3a%2f%2fwww.merchant.reduniq.pt%2ferror%2f**  
 &NOTIFICATIONURL=https%3a%2f%2fwww.merchant.pt%2fnotification%2f  
 &PRIVATEDATA\_NAME\_0=key+%231  
 &PRIVATEDATA\_VALUE\_0=value+%231  
 &PRIVATEDATA\_NAME\_1=key+%232  
 &PRIVATEDATA\_VALUE\_1=value+%232  
 &PRIVATEDATA\_NAME\_2=key+%233  
 &PRIVATEDATA\_VALUE\_2=value+%233  
 &LANGUAGECODE=por

#### Estrutura de dados da Resposta:

Campo	Descrição	Formato	Observações
result.code	Código do resultado: 00000000 - OK	AN8	Ver em anexo "Códigos e mensagens"
result.message	Mensagem do resultado	AN255	-
token	token atribuído ao pagamento	AN50	-
redirectUrlt	URL para redirecionar o cliente para a página de pagamento	AN255	-

#### Exemplo de uma resposta em formato SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:reduniqPaymentGateway">
  <SOAP-ENV:Body>
    <ns1:initPaymentResponse xmlns:ns1="urn:reduniqPaymentGateway">
      <result xsi:type="tns:resultType">
        <code xsi:type="xsd:string">00000000</code>
        <message xsi:type="xsd:string"></message>
      </result>
      <token xsi:type="xsd:string">f34b971751d47fcb418dfd590d24c124</token>
      <redirectUrl xsi:type="xsd:string">https://www.xpto.com/token</redirectUrl>
    </ns1:initPaymentResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Exemplo de uma resposta em formato NVP:

```
RESULT_CODE=00000000
&RESULT_MESSAGE=
&TOKEN=db3df58480703d9d91fac1f91f9d0ef7
&REDIRECTURL=https%3A%2F%2Fwww.xpto.com%2Ftoken%2F
```

### getResult

A função “*getResult*” permite ao comerciante consultar os detalhes e o resultado de um pagamento, indicando no pedido, o token associado ao pagamento.

### Estrutura de dados do pedido:

Campo	Descrição	Obrig.	Formato	Observações
api.username	Username da API	sim	AN100	-
api.password	Password da API	sim	AN20	-
token	token atribuído ao pagamento	sim	AN50	-

### Exemplo de um pedido em formato SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:reduniqPaymentGateway">
  <SOAP-ENV:Body>
    <tns:getResult xmlns:tns="urn:reduniqPaymentGateway">
      <api xsi:type="tns:apiType">
        <username xsi:type="xsd:string">demouser</username>
        <password xsi:type="xsd:string">abc123</password>
      </api>
      <token xsi:type="xsd:string">01417b2b102624f00fcd9a0226820601</token>
    </tns:getResult>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Exemplo de um pedido em formato NVP:

```
METHOD=getResult
&API_USERNAME=demouser
&API_PASSWORD=abc123
&TOKEN=01417b2b102624f00fcd9a0226820601
```



**Estrutura de dados da resposta:**

Campo	Descrição	Formato	Observações
result.code	Código do resultado: 10000000 – OK (RU E-Com) 20000000 – OK (PayPal) 30000000 – OK (AMEX Web2Pay) 40000000 – OK (SIBS MB WAY)	AN8	-
result.message	Mensagem do resultado	AN255	-
transaction.id	ID único da transação	AN50	-
transaction.isFraud	-	AN1	Disponível para o REDUNIQ Ecommerce 0 – risco de fraude não detectado 1 – risco de fraude
transaction.date	Data e hora da transação	AN19	2012-12-15 20:30:00
payment.solution	Solução de pagamento selecionada pelo cliente	N3	100 – REDUNIQ E-Commerce 101 – PayPal 102 – AMEX Web2Pay 103 – SIBS MB WAY
payment.amount	Valor total do pagamento	N12	O valor 100 corresponde a 1€ O valor 1599 corresponde a 15,99€
payment.action	Código da ação de pagamento	N4	100 – Autorização 101 – Autorização + Captura
privateData	Dados privados	-	Ver estrutura de dados "privateData"

**Exemplo de uma resposta em formato SOAP:**

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:redunIQPaymentGateway">
  <SOAP-ENV:Body>
    <ns1:getResultResponse xmlns:ns1="urn:redunIQPaymentGateway">
      <result xsi:type="tns:resultType">
        <code xsi:type="xsd:string">20000000</code>
        <message xsi:type="xsd:string">Success</message>
      </result>
      <transaction xsi:type="tns:transactionType">
        <id xsi:type="xsd:string">9JL84341LL847331L</id>
        <isFraud xsi:nil="true" xsi:type="xsd:string"/>
        <date xsi:type="xsd:string">2013-01-27 23:15:57</date>
      </transaction>
      <payment xsi:type="tns:paymentType">
        <amount xsi:type="xsd:integer">12500</amount>
        <action xsi:type="xsd:integer">100</action>
        <solution xsi:type="xsd:integer">101</solution>
      </payment>
      <privateData xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="tns:privateDataItem[3]">
        <item xsi:type="tns:privateDataItem">
          <name xsi:type="xsd:string">key #1</name>
```

```

    <value xsi:type="xsd:string">value #1</value>
  </item>
  <item xsi:type="tns:privateDataItem">
    <name xsi:type="xsd:string">key #2</name>
    <value xsi:type="xsd:string">value #2</value>
  </item><item xsi:type="tns:privateDataItem">
    <name xsi:type="xsd:string">key #3</name>
    <value xsi:type="xsd:string">value #3</value>
  </item>
</privateData>
</ns1:getResultResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Exemplo de uma resposta em formato NVP:

```

RESULT_CODE=20000000
&RESULT_MESSAGE=Success
&TRANSACTION_ID=9JL84341LL847331L
&TRANSACTION_ISFRAUD=
&TRANSACTION_DATE=2013-01-27+23%3A15%3A57
&PAYMENT_AMOUNT=12500
&PAYMENT_ACTION=100
&PAYMENT_SOLUTION=101
&PRIVATEDATA_NAME_0=key+%231
&PRIVATEDATA_VALUE_0=value+%231
&PRIVATEDATA_NAME_1=key+%232
&PRIVATEDATA_VALUE_1=value+%232
&PRIVATEDATA_NAME_2=key+%233
&PRIVATEDATA_VALUE_2=value+%233

```

### doCapture

A função “doCapture” permite ao comerciante aceitar uma autorização de pagamento anteriormente validada. Estrutura de dados do pedido:

Campo	Descrição	Obrig.	Formato	Observações
api.username	Username da API	sim	AN100	-
api.password	Password da API	sim	AN20	-
transaction.id	ID único da transação	sim	N50	Devolvido pela função “getResult”
payment.amount	Valor da transação	sim	N12	O valor 100 corresponde a 1€ O valor 1599 corresponde a 15,99€
payment.action	Código da ação de pagamento	sim	N3	200 Total (Full) 201 Parcial (Partial)

**Exemplo de um pedido em formato SOAP:**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:reduniqPaymentGateway">
  <SOAP-ENV:Body>
    <tns:doCapture xmlns:tns="urn:reduniqPaymentGateway">
      <api xsi:type="tns:apiType">
        <username xsi:type="xsd:string">demouser</username>
        <password xsi:type="xsd:string">abc123</password>
      </api>
      <transaction xsi:type="tns:transactionType">
        <id xsi:type="xsd:string">9JL84341LL847331L</id>
      </transaction>
      <payment xsi:type="tns:paymentType">
        <amount xsi:type="xsd:integer">12500</amount>
        <action xsi:type="xsd:integer">200</action>
      </payment>
    </tns:doCapture>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Exemplo de um pedido em formato NVP:**

```
METHOD=doCapture
&API_USERNAME=demouser
&API_PASSWORD=abc123
&TRANSACTION_ID=9JL84341LL847331L
&PAYMENT_AMOUNT=12500
&PAYMENT_ACTION=200
```

**Estrutura de dados da resposta:**

Campo	Descrição	Formato	Observações
result.code	Código do resultado:	AN8	Ver em anexo "Códigos e mensagens"
result.message	Mensagem do resultado	AN255	-
transaction.id	ID único da transação	AN50	-
transaction.date	Data e hora da transação	AN16	2012-12-15 20:30:00

**Exemplo de uma resposta em formato SOAP:**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:reduniqPaymentGateway">
  <SOAP-ENV:Body>
    <ns1:doCaptureResponse xmlns:ns1="urn:reduniqPaymentGateway">
```

```

<result xsi:type="tns:resultType">
  <code xsi:type="xsd:string">20000000</code>
  <message xsi:type="xsd:string">Success</message>
</result>
<transaction xsi:type="tns:transactionType">
  <id xsi:type="xsd:string">1UR62345LF0480138</id>
  <date xsi:type="xsd:string">2013-01-27 23:36:31</date>
</transaction>
</ns1:doCaptureResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Exemplo de uma resposta em formato NVP:

```

RESULT_CODE=20000000
&RESULT_MESSAGE=Success
&TRANSACTION_ID=1UR62345LF0480138
&TRANSACTION_DATE=2013-01-27+23%3A36%3A31

```

### doRefund

A função “doRefund” permite ao comerciante reembolsar um pagamento autorizado e confirmado.  
Estrutura de dados do pedido:

Campo	Descrição	Obrig.	Formato	Observações
api.username	Username da API	sim	AN100	-
api.password	Password da API	sim	AN20	-
transaction.id	ID único da transação	sim	N50	Devolvido pela função “getResult”
payment.amount	Valor da transação	sim	N12	-
payment.action	Código da ação de pagamento	sim	N4	300 Reembolso (Refund)
comment	Comentário / observação	não	AN255	-

### Exemplo de um pedido em formato SOAP:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:reduniqPaymentGateway">
  <SOAP-ENV:Body>
    <tns:doRefund xmlns:tns="urn:reduniqPaymentGateway">
      <api xsi:type="tns:apiType">
        <username xsi:type="xsd:string">demouser</username>
        <password xsi:type="xsd:string">abc123</password>
      </api>
      <transaction xsi:type="tns:transactionType">
        <id xsi:type="xsd:string">74G405248W656331Y</id>

```

```

</transaction>
<payment xsi:type="tns:paymentType">
  <amount xsi:type="xsd:integer">12500</amount>
  <action xsi:type="xsd:integer">300</action>
</payment>
<comment xsi:type="xsd:string">devolvido</comment>
</tns:doRefund>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Exemplo de um pedido em formato NVP:

```

METHOD=doRefund
&API_USERNAME=demouser
&API_PASSWORD=abc123
&TRANSACTION_ID=74G405248W656331Y
&PAYMENT_AMOUNT=12500
&PAYMENT_ACTION=300
&COMMENT=devolvido

```

### Estrutura de dados da resposta:

Campo	Descrição	Formato	Observações
result.code	Código do resultado:	AN8	Ver em anexo "Códigos e mensagens"
result.message	Mensagem do resultado	AN255	-
transaction.id	ID único da transação	AN50	-
transaction.date	Data e hora da transação	AN16	2012-12-15 20:30:00

### Exemplo de uma resposta em formato SOAP:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:reduniqPaymentGateway">
  <SOAP-ENV:Body>
    <ns1:doRefundResponse xmlns:ns1="urn:reduniqPaymentGateway">
      <result xsi:type="tns:resultType">
        <code xsi:type="xsd:string">20000000</code>
        <message xsi:type="xsd:string">Success</message>
      </result>
      <transaction xsi:type="tns:transactionType"></transaction>
    </ns1:doRefundResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

**Exemplo de uma resposta em formato NVP:**

RESULT\_CODE=20000000  
 RESULT\_MESSAGE=Success  
 TRANSACTION\_ID=  
 TRANSACTION\_DATE=

**doVoid**

A função “doVoid”, dependendo do meio de pagamento utilizado, permite:

- PayPal: Cancelar um pagamento autorizado;
- SIBS MB WAY: Anular uma outra operação financeira submetida;

**Estrutura de dados do pedido:**

Campo	Descrição	Obrig.	Formato	Observações
api.username	Username da API	sim	AN100	-
api.password	Password da API	sim	AN20	-
transaction.id	ID único da transação	sim	N50	Devolvido pela última operação

**Exemplo de um pedido em formato SOAP:**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:reduniqPaymentGateway">
  <SOAP-ENV:Body>
    <tns:doVoid xmlns:tns="urn:reduniqPaymentGateway">
      <api xsi:type="tns:apiType">
        <username xsi:type="xsd:string">demouser</username>
        <password xsi:type="xsd:string">abc123</password>
      </api>
      <transaction xsi:type="tns:transactionType">
        <id xsi:type="xsd:string">1Y763144JD355390R</id>
      </transaction>
    </tns:doVoid>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Exemplo de um pedido em formato NVP:**

METHOD=doVoid  
 API\_USERNAME=demouser  
 API\_PASSWORD=abc123  
 TRANSACTION\_ID=1Y763144JD355390R

**Estrutura de dados da resposta:**

Campo	Descrição	Formato	Observações
result.code	Código do resultado:	AN8	Ver em anexo "Códigos e mensagens"
result.message	Mensagem do resultado	AN255	-
transaction.id	ID único da transação	AN50	-
transaction.date	Data e hora da transação	AN16	2012-12-15 20:30:00

**Exemplo de uma resposta em formato SOAP:**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:reduniqPaymentGateway">
  <SOAP-ENV:Body>
    <ns1:doVoidResponse xmlns:ns1="urn:reduniqPaymentGateway">
      <result xsi:type="tns:resultType">
        <code xsi:type="xsd:string">20000000</code>
        <message xsi:type="xsd:string">Success</message>
      </result>
      <transaction xsi:type="tns:transactionType">
        <id xsi:type="xsd:string">1Y763144JD355390R </id>
        <date xsi:type="xsd:string">2013-01-27 23:36:31</date>
      </transaction>
    </ns1:doVoidResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Exemplo de uma resposta em formato NVP:**

```
RESULT_CODE=20000000
&RESULT_MESSAGE=Success
&TRANSACTION_ID=1Y763144JD355390R
&TRANSACTION_DATE=2013-01-27+23%3A36%3A31
```

# Anexos

## Códigos e mensagens

### Tipos de código de resposta:

Código	Descrição
0xxxxxx	Códigos de resposta do <i>gateway de pagamentos</i>
1xxxxxx	Códigos de resposta do REDUNIQ E-Commerce
2xxxxxx	Códigos de resposta do PayPal
3xxxxxx	Códigos de resposta do AMEX Web2Pay
4xxxxxx	Códigos de resposta do SIBS MB WAY

### Códigos de resposta do *gateway de pagamentos*

Código	Mensagem
100001	Authentication failed
100002	This merchant has no associated payment solutions
100003	Access / Authentication failed: doWebPayment (Reduniq E-commerce)
100004	Access / Authentication failed: SetExpressCheckout (PayPal)
100005	Access / Authentication failed: getWebPaymentDetails (Reduniq E-commerce)
100006	Access / Authentication failed: DoExpressCheckoutPayment (PayPal)
100007	Invalid token
100008	The payment process has not yet started
100009	The payment process is in progress
100010	Invalid transaction
100011	Access / Authentication failed: doCapture (Reduniq E-commerce)
100012	Access / Authentication failed: doRefund (PayPal)
100013	Access / Authentication failed: GetExpressCheckoutDetails (PayPal)
100014	Access / Authentication failed: doCapture (PayPal)
100015	Access / Authentication failed: doRefund (Reduniq E-commerce)
100016	Service not implemented for this payment solution
100017	Invalid service
100018	Access / Authentication failed: RequestNoCardRead (AMEX Web2Pay)
100019	Access / Authentication failed: RequestAuthorised (AMEX Web2Pay)
100020	Declined - Reversed / CVV Code non match (AMEX Web2Pay)
100021	Declined - Not Reversed / CVV Code non match (AMEX Web2Pay)
100022	Access / Authentication failed: ReverseByTxID (AMEX Web2Pay)
100023	Access / Authentication failed: FinancialOperationRequest/022-024 (MB WAY)
100024	Access / Authentication failed: FinancialOperationResult (MB WAY)
100025	Access / Authentication failed: FinancialOperationRequest/025 (MB WAY)
100026	Access / Authentication failed: FinancialOperationRequest/023 (MB WAY)
100027	Access / Authentication failed: FinancialOperationRequest/048 (MB WAY)
200001	Internal error: Error reading information from the merchant
200002	Internal error: Error creating transaction
200003	Internal error: Error reading the transaction result



300000	Invalid parameter: payment.amount
300001	Invalid parameter: payment.action
300002	Invalid parameter: payment.description
300003	Invalid parameter: order.ref
300004	Invalid parameter: order.amount
300005	Invalid parameter: order.taxes
300006	Invalid parameter: order.date
300007	Invalid parameter: buyer.firstName
300008	Invalid parameter: buyer.lastName
300009	Invalid parameter: buyer.email
300010	Invalid parameter: buyer.shipping.name
300011	Invalid parameter: buyer.shipping.street1
300012	Invalid parameter: buyer.shipping.street2
300013	Invalid parameter: buyer.shipping.city
300014	Invalid parameter: buyer.shipping.zipCode
300015	Invalid parameter: buyer.shipping.country
300016	Invalid parameter: buyer.shipping.phone
300017	Invalid parameter: returnUrlOk
300018	Invalid parameter: returnUrlError
300019	Invalid parameter: notificationUrl
300020	Invalid parameter: languageCode
300021	Invalid parameter: transaction.id
300022	Invalid parameter: comment
300023	Invalid parameter: buyer.shipping.amount
300024	Invalid parameter: payment.amount (do not match order amount + order taxes + shipping amount)
300025	Invalid parameter: order.shipping
300026	Invalid parameter: buyer.shipping.state
300027	Invalid parameter: order.shipping (shipping = 0 and shipping.amount > 0)
300028	Invalid parameter: buyer.shipping (name, street1, city, zipCode and country are mandatory if shipping=1)
400000	Invalid parameter: order.details.name
400001	Invalid parameter: order.details.amount
400002	Invalid parameter: order.details.quantity
400003	Invalid parameter: order.details.tax
400004	Invalid parameter: order.details (exceeds the limit of 100 items)
400005	Invalid parameter: order.details (item amounts do not match order amounts + order taxes)
400006	Invalid parameter: order.details.category
500000	Invalid parameter: privateData.key
500001	Invalid parameter: privateData.value
500002	Invalid parameter: privateData (exceeds the limit of 100 items)
600000	Payment canceled by the user
700000	Timeout: Redirect to the payment page
700001	Timeout: In selecting payment method
700002	Timeout: Finalizing payment

## Gestão de timeouts

O gateway de pagamentos implementa três timeouts:

- . Após a chamada da função “initPayment”, o token devolvido é válido por um período de 60 segundos. Ou seja, o tempo decorrido entre a criação do token e o redireccionamento do cliente para a página do gateway, não pode exceder os 60 segundos.
- . Na página do gateway, o cliente dispõe de 15 minutos para seleccionar o modo de pagamento pretendido.
- . Todo o processo de pagamento não deve exceder os 45 minutos. Cada solução de pagamentos implementa os seus próprios timeouts. Para mais informações, consultar os respectivos guias de integração.

Empowered by  
**UNICRE**