



# Criação da Transação - Bcash - API

## Conteúdo

1. Descrição do Serviço.....	3
2. Processo.....	5
3. Autenticação.....	6
4. Parâmetros de envio.....	7
5. Retorno.....	16
5.1. HTTP Code.....	16
5.2. Retorno em caso de sucesso.....	17
5.2.1. Informações retornadas.....	17
5.2.2. Exemplos de retorno.....	18
5.3. Retorno em caso de erro.....	19
5.3.1. Informações retornadas de erro.....	19
5.4. Mensagens de Erro.....	19
5.5 Exemplos de retorno.....	25
6. Exemplo de codificação.....	26

## 1. Descrição do Serviço

A criação da transação tem como finalidade o recebimento de dados referentes a uma transação de compra e venda e também a execução dos procedimentos para o seu processamento no Bcash e nas empresas parceiras (análise de risco, bancos, processadoras de cartão de crédito, etc.).

As formas de pagamento disponíveis são: boleto, transferência online e cartão de crédito.

Confira o processo para o lojista iniciar a utilização deste serviço:

### Cliente solicita liberação das APIs



- Libera acesso
- Libera aba no Banking
- Gera chave de identificação (token do cliente)

**Avisa usuário**

O processo resumido para integração com o serviço, enviando e processando as transações, é exibido abaixo:

### Loja envia requisição



- Valida acesso à API
- Valida autenticidade
- Valida dados de requisição
- Processa requisição

**Loja recebe resposta**

Nesta integração, o lojista faz uma requisição padronizada via POST e recebe as informações de processamento de acordo com os dados enviados.

Abaixo, é mostrado um exemplo da requisição:

### Exemplo:



- ➡ A resposta da requisição poderá ser: não autorizado, erro ou sucesso:
- ➡ Não autorizado: O cliente não está autorizado a utilizar a API (401)
- ➡ Erro: A requisição não foi enviada corretamente ou o acesso foi validado, mas a requisição apresenta algum erro nos parâmetros enviados ou o serviço está com alguma indisponibilidade ou problemas técnicos no momento (400, 405, 415, 500 e 503)

Sucesso: Acesso validado e requisição processada com sucesso (200)



Não autorizado: 401

Erro: 400, 405, 415, 500 e 503

Sucesso: 200

## 2. Processo

O processo de criação de transação deverá seguir os seguintes passos:

- Cliente envia POST, conforme parâmetros dos itens 3 e 4 desta documentação.
- A solicitação é recebida.
- O usuário é autenticado conforme informações enviadas no header da requisição.
- A transação é criada conforme os parâmetros enviados na requisição.
- O cliente recebe uma mensagem de sucesso ou erro, formatada como XML ou JSON (com encode HTML), conforme o parâmetro “typeReturn” na URL e com a codificação do parâmetro “encode”.

### 3. Autenticação – Padrão OAuth

Para autenticação, o serviço utiliza o padrão aberto para autorização OAuth (Open Authorization). Neste modelo, os dados são enviados no cabeçalho de uma solicitação para garantir que a requisição seja autêntica. A loja envia suas transações com este cabeçalho e as informações são validadas.

Serviço disponível através das URL's abaixo:

<https://api.bcash.com.br/service/createTransaction/xml/>

<https://api.bcash.com.br/service/createTransaction/json/>

Tabela 1- Dados para autenticação

Nome	Descrição
Realm	URL do serviço. Descreve o serviço que será consumido e o tipo de retorno da requisição.
oauth_consumer_key	Chave identificadora do cliente gerada pelo Bcash.
oauth_nonce	String única gerada para cada requisição. O “nonce” não permite que uma mesma requisição seja postada novamente.
oauth_signature	Armazena os parâmetros do cabeçalho codificados em Base64. É composto dos campos oauth_consumer_key, oauth_signature_method, oauth_timestamp, oauth_nonce, e oauth_version.
oauth_signature_method	Método de criptografia utilizado para gerar o oauth_signature no nosso caso apenas PLAINTEXT (codificado em Base64).
oauth_timestamp	Número de segundos desde 1 de Janeiro de 1970 00:00:00 GMT. O valor do “timestamp” deve ser um inteiro positivo e deve ser igual ou maior que o timestamp utilizado em requisições anteriores para o mesmo cliente.
oauth_version	Campo é opcional que indica a versão do OAuth. Esta documentação baseia-se na versão 1.0.

Siga estes passos para montar o modelo de autorização OAuth:

- Inserir os parâmetros na requisição: Authorization e Content-Type.
- O parâmetro Authorization é composto por: OAuth realm=, oauth\_consumer\_key=, oauth\_nonce, oauth\_signature=, oauth\_signature\_method=, oauth\_timestamp, oauth\_version=
- O parâmetro oauth\_signature é criado por 4 passos:

**1) Coletar os parâmetros oauth\_consumer\_key, oauth\_nonce, oauth\_signature\_method, oauth\_timestamp e oauth\_version.**

**2) Codificar os parâmetros com URL encoding (Percent-encode) e ordenar primeiro pelo nome e depois pelo valor. Na implementação do OAuth, somente pelo nome é suficiente. Desta forma, os parâmetros devem ser informados na ordem em que foram mencionados acima.**

**3) Os parâmetros devem ser “concatenados” em uma string separados por “&”.**

Exemplo: oauth\_consumer\_key=515w1f51f515f665f5ewf&oauth\_nonce=wef8834t54451t...

Portanto, o resultado final deve ser uma string semelhante a esta:

oauth\_consumer\_key=a1191fd420e0164c2f9aeac32ed35d23&

oauth\_nonce=109843dea839120a&

oauth\_signature\_method=PLAINTEXT&

oauth\_timestamp=1258308730&

oauth\_version=1.0

**4) Aplicar o valor final (string acima resultante) à função base64\_encode**

**Exemplo:**

Authorization: OAuth

```
realm="https://api.bcash.com.br/service/createTransaction/xml/",  
  oauth_consumer_key=a1191fd420e0164c2f9aeac32ed35d23,  
  oauth_nonce=85a55859fde262ba,  
  oauth_signature=d8e19bb988110380a72f6dba33b2ba5903272fe1,  
  oauth_signature_method=HMAC-SHA1,  
  oauth_timestamp=1258308689,  
  oauth_version=1.0
```

## 4. Parâmetros de envio

Serviço disponível em:

[https://api.bcash.com.br/service/createTransaction/\[tipo do retorno\]](https://api.bcash.com.br/service/createTransaction/[tipo do retorno])

O tipo de retorno da URL acima poderá ser XML ou JSON. As informações da requisição, presentes no parâmetro “data”, seguem o formato JSON com o conteúdo “escapado” para encode de URI (escape HTML).

No POST enviado, os seguintes parâmetros deverão ser informados:

Nome	Obrigatório	Tamanho	Tipo	Descrição
data	Sim	X	String	Objeto de requisição (TransactionRequest) no padrão JSON preenchido. Deverá conter as informações para realização do pedido descritas na tabela 2.
encode	Não	50	String	Codificação de resposta: “UTF-8” ou “ISO-8859-1”. Caso não seja enviado, o <i>default</i> é “UTF-8”.
version	Não	10	String	Versão da API de consulta disponível consumida. Ex.: 1.0. Caso não seja enviada, será considerada a versão mais recente da API disponibilizada pelo Bcash.

**Tabela 2 – Parâmetros de entrada da criação**

O objeto JSON **TransactionRequest** é formado pelos seguintes elementos:

sellerMail, orderId, buyer (objeto JSON do tipo CustomerRequest), free, freight, freightType, discount, addition, paymentMethod (objeto JSON do tipo PaymentMethodRequest), urlReturn, urlNotification, products (lista de objetos JSON do tipo ProductRequest), dependentTransactions (lista de objetos JSON do tipo DependentTransactionRequest), installments, creditCard (objeto JSON do tipo CreditCardRequest), currency, acceptedContract, viewedContract, campaigned



O objeto JSON **CustomerRequest** é formado pelos elementos:

mail, name, cpf, phone, cellPhone, address (objeto JSON do tipo AddressRequest), gender, birthDate, rg, issueRgDate, organConsignorRg, stateConsignorRg, companyName, cnpj, searchToken

O objeto JSON **AddressRequest** é formado pelos seguintes elementos:

address, number, complement, neighborhood, city, state, zipCode

O objeto JSON **PaymentMethodRequest** é formado pelos seguintes elementos:

code

O objeto JSON **ProductRequest** é formado pelos seguintes elementos:

code, description, amount, value, extraDescription

O objeto JSON **DependentTransactionRequest** é formado pelos seguintes elementos:

email, value

O objeto JSON **CreditCardRequest** é formado pelos seguintes elementos:

holder, number, securityCode, maturityMonth, maturityYear

Confira mais detalhes sobre cada um dos atributos mencionados acima:

Nome	Obrig.	Tam.	Tipo	Descrição
sellerMail	Sim	80	String	E-mail da loja cadastrada no Bcash. Ex.: empresa@provedor.com
orderId	Não	50	String	Código do pedido atribuído pela loja. Ex.: 123
buyer	X	X	Objeto	Objeto JSON do tipo CustomerRequest.

(CustomerRequest) mail	Sim	80	String	E-mail do comprador. Ex.: cliente@provedor.com
(CustomerRequest) name	Sim	80	String	Nome do comprador. Ex.: Antonio Junior
(CustomerRequest) cpf	Sim	17	String	CPF do comprador. Ex.: 999999999999
(CustomerRequest) phone	Sim*	20	String	Telefone do comprador: DDD + número telefônico  O número do DDD deve conter 2 dígitos numéricos. Ex: 14  O número do telefone deve conter exatamente 8 dígitos numéricos. Ex: 1432218331  O primeiro dígito do número do telefone precisa ser maior ou igual a 2 e menor que 7. Ex: 32218331, 65547132
(CustomerRequest) cellPhone	Sim**	20	String	Celular: DDD + número telefônico.  O número do DDD deve conter 2 dígitos numéricos. Ex: 14  Se o número de DDD for igual a 11, o número de telefone pode conter 8 ou 9 dígitos numéricos. Ex: 1199810400, 11999810400  Se o DDD for igual a 11 o primeiro dígito pode ser 5, 7, 8 ou 9.  Se o DDD for diferente de 11 o primeiro dígito pode ser 7, 8 ou 9.
(CustomerRequest) address	X	X	Objeto	Objeto JSON do tipo AddressRequest.
(CustomerRequest) (AddressRequest) address	Sim***	100	String	Endereço do comprador. Ex.: Av. Tiradentes
(CustomerRequest)	Sim***	10	String	Número do endereço do comprador.

(AddressRequest) number				Ex.: 1254
(CustomerRequest) (AddressRequest) complement	Não	80	String	Complemento do endereço do comprador. Ex.: Apartamento 10
(CustomerRequest) (AddressRequest) neighborhood	Sim***	50	String	Bairro do comprador. Ex.: Centro
(CustomerRequest) (AddressRequest)city	Sim***	255	String	Cidade do comprador. Ex.: São Paulo
(CustomerRequest) (AddressRequest) state	Sim***	2	String	Estado do comprador. Ex.: MG
(CustomerRequest) (AddressRequest) zipCode	Sim***	9	String	CEP do comprador. Ex.: 17500000
(CustomerRequest) gender	Sim	1	String	Sexo do comprador. Ex.: M
(CustomerRequest) birthDate	Não	10	String	Data de nascimento do comprador. No formato dd/MM/aaaa. Ex.: 26/12/1985
(CustomerRequest) rg	Não	17	String	RG do comprador. Ex.: 99999999
(CustomerRequest) issueRgDate	Não	10	String	Data da emissão do RG do comprador. No formato dd/MM/aaaa. Ex.: 15/08/1998
(CustomerRequest) organConsignorRg	Não	15	String	Órgão emissor do RG do comprador. Ex.: SSP
(CustomerRequest) stateConsignorRg	Não	2	String	Estado emissor do RG do comprador. Ex.: MG
(CustomerRequest) companyName	Não	255	String	Razão social do comprador. Enviado somente quando a conta do cliente for conta Jurídica. Ex.: Razão Social da minha empresa

(CustomerRequest) cnpj	Não	30	String	CNPJ do comprador. Enviado somente quando a conta do cliente for conta Jurídica.  Ex.: 99999999999999
(CustomerRequest) searchToken	Não	X	String	Token gerado a partir do serviço de consulta de conta.  Ex.: P2Z/G54BkrkM7gs+TlkdlQ==
free	Não	255	String	Campo de livre digitação. Pode ser utilizado para algum parâmetro adicional de identificação da venda.
freight	Não	(8,2)	Decimal	Valor do frete.  Ex.: 10.25
freightType	Não	80	String	Tipo do frete (SEDEX, PAC, ENCOMENDA, CARTA_REGISTRADA e E_SEDEX).  Ex.: SEDEX
discount	Não	(8,2)	Decimal	Valor total do desconto atribuído pela loja.  Ex.: 2.25
addition	Não	(8,2)	Decimal	Valor total do acréscimo feito pela loja.  Ex.: 2.25
paymentMethod	X	X	Objeto	Objeto JSON do tipo PaymentMethodRequest.
(PaymentMethodRequest) code	Sim	3	Int	Código do meio de pagamento utilizado para processar a transação.  Boleto = 10 Transferência Online Banco do Brasil = 58 Transferência Online Bradesco = 59 Transferência Online Itaú = 60 Transferência Online Banrisul = 61 Transferência Online HSBC = 62 Cartão de Crédito: VISA = 1 MASTERCARD = 2 AMERICAN_EXPRESS = 37 AURA = 45

				DINERS = 55 HIPERCARD = 56 ELO = 63  Ex.: 10 (Boleto)
urlReturn	Não	255	String	URL que direciona o usuário para a loja. Ex.: <a href="http://www.bcash.com.br/loja/retorno.php">http://www.bcash.com.br/loja/retorno.php</a>
urlNotification	Não	255	String	URL que a loja irá receber as informações de alteração de status da transação. Ex.: <a href="http://www.bcash.com.br/loja/aviso.php">http://www.bcash.com.br/loja/aviso.php</a>
products	X	X	List	Lista de objetos JSON do tipo ProductRequest.
product	X	X	Objeto	objeto JSON do tipo ProductRequest
(ProductRequest) code	Sim	50	String	Código que identifica o produto em sua loja Ex.: 446
(ProductRequest) extendedWarranty	X	X	Objeto	objeto JSON do tipo ExtendedWarrantyRequest
(ProductRequest) (ExtendedWarrantyRequest) monthWarranty	Não	4	Int	Meses de Garantia ofertados pelo próprio fabricante ou lojista (após esses meses a garantia estendida passa a contar). Ex.: 6
(ProductRequest) (ExtendedWarrantyRequest) amount	Não	X	Int	Quantidade de garantias correspondente a quantidade de produtos comprada (por exemplo se são 3 produtos X comprados o número de garantias deve ser menor ou igual a 3). Ex.: 2
(ProductRequest) (ExtendedWarrantyRequest) token	Não	X	String	Token retornado no serviço de consulta de garantia estendida, disponibilizado pelo Bcash no endereço: <a href="https://api.bcash.com.br/service/searchValueWarranty/xml/">https://api.bcash.com.br/service/searchValueWarranty/xml/</a> O token deve ser enviado para cada tipo de produto que tenha garantia. Ex.: wPCVaT8v3Q3wylAV6OGpBD7eN7LZBnvj4F1M/626+Y2c1BJP4EG4lalwZP8LmYJJ49ielxidxFY+mBdKVgbcBg==

(ProductRequest) description	Sim	255	String	Descrição do(s) produto(s) comprado(s). Será visualizada pelo cliente no Bcash. Ex.: Camiseta da seleção brasileira
(ProductRequest) amount	Sim	X	Int	Quantidade comprada do(s) produto(s). Ex.: 2
(ProductRequest) value	Sim	(8,2)	Decimal	Valor unitário do produto. Ex.: 10.95
(ProductRequest) extraDescription	Não	255	String	Descrição adicional do produto. Ex.: Tamanho P
dependentTransactions	X	X	List	Lista de objetos JSON do tipo DependentTransactionRequest
dependentTransaction	X	X	Objeto	Objeto JSON do tipo DependentTransactionRequest
(DependentTransactionRequest) email	Não	80	String	Email da conta dependente. Ex.: empresa@provedor.com
(DependentTransactionRequest) value	Não	(8,2)	Decimal	Valor monetário da transação dependente. Ex.: 10.95
installments	Não	X	Int	Quantidade de parcelas que a compra será processada. Ex.: 5
creditCard	X	X	Objeto	Objeto JSON do tipo CreditCardRequest.
(CreditCardRequest) holder	Sim****	100	String	Nome do titular do cartão de crédito. Ex.: JOAO D S SILVA
(CreditCardRequest) number	Sim****	30	String	Número do cartão de crédito do cliente. Ex.: 4111111111111110
(CreditCardRequest) securityCode	Sim****	6	String	Código de segurança do cartão de crédito, geralmente encontra-se no verso do cartão. Ex.: 123

(CreditCardRequest) maturityMonth	Sim****	2	String	Mês de vencimento do cartão de crédito. Ex.: 08
(CreditCardRequest) maturityYear	Sim****	4	String	Ano de vencimento do cartão de crédito. Ex.: 2017
currency	Não	3	String	Moeda utilizada para a transação. Ex.: BRL(Real)
acceptedContract	Sim	1	String	Parâmetro utilizado para informar se o comprador aceitou os termos do contrato (S – Sim ou N - Não). Ex.: S
viewedContract	Sim	1	String	Parâmetro utilizado para informar se o comprador visualizou os termos do contrato (S – Sim ou N - Não). Ex.: S
campaignId	Não	255	String	Identificador da campanha da loja no Bcash.

\* Campo Obrigatório, caso não seja informado o campo cellPhone.

\*\* Campo Obrigatório, caso não seja informado o campo phone.

\*\*\* O campo de endereço deixa de ser obrigatório para vendedores configurados para “Checkout de Serviços”.

\*\*\*\* Campo obrigatório somente para a escolha de meio de pagamento do tipo “Cartão de Crédito”.

### Exemplo de JSON de requisição TransactionRequest:

JSON:

JSON:

```
{
  "products": [
    {
      "code": "1236456",
      "description": "Produto de teste",
      "amount": "5",
      "value": "15.25",
      "extraDescription": "Este produto é um exemplo",
      "extendedWarranty": {
        "monthWarranty": "6",
        "originalSafePrice": "",
        "buscapeld": "",
        "categoryBuscape": "",
        "extendedWarrantyPeriod": "",
        "amount": "2",
        "brand": "",
        "token": "wPCVaT8v3Q3wylAV6OGpBD7eN7LZBnvj4F1M/626+Y2c1BJP4EG4lalwZP8LmYJJ49ielxidxFY+mBdKVgbCBg=="
      },
      "dependentTransactions": [],
      "buyer": {
        "address": {
          "address": "Rua teste teste",
          "number": "123",
          "complement": "apto. 000",
          "neighborhood": "Bairro de exemplo",
          "city": "Exemplo de cidade",
          "state": "MG",
          "zipCode": "12345-789",
          "mail": "comprador@comprador.com",
          "name": "Comprador Teste",
          "cpf": "850-822-365-04",
          "phone": "34-3311-9999",
          "cellPhone": "34-9999-1111",
          "gender": "M",
          "birthDate": "01/01/1970",
          "rg": "11337733",
          "issueRgDate": "01/01/1990",
          "organConsignorRg": "SSP",
          "stateConsignorRg": "MG",
          "companyName": "Empresa de teste",
          "cnpj": "72-139-715/0001-30",
          "searchToken": "",
          "ipAddress": "162.198.016.121",
          "paymentMethod": {
            "code": "58",
            "creditCard": null,
            "sellerMail": "lojamodelo@pagamentodigital.com.br",
            "idTransaction": "",
            "transactionKey": "",
            "origin": "P",
            "orderId": "123546",
            "free": "Abc",
            "def": "freight": "10.50",
            "freightType": "SEDEX",
            "discount": "1.95",
            "addition": "2.65",
            "urlReturn": "retorno.bcash.com.br",
            "urlNotification": "aviso.bcash.com.br",
            "installments": "1",
            "campaignId": "",
            "platformId": "",
            "currency": "BRL",
            "lomadee": "1",
            "idAdvertiser": "22",
            "acceptedContract": "S",
            "viewedContract": "S"
          }
        }
      }
    }
  ]
}
```

## 5. Retorno

Este serviço possui dois formatos de retornos previstos: um no caso de sucesso e outro no caso de erro. Ambos respeitam a escolha do tipo do retorno (“XML” ou “JSON”) e a codificação (“UTF-8” ou “ISO-8859-1”).

### 5.1. HTTP Code

O Bcash permite que o cliente identifique, através da resposta do protocolo HTTP retornada, se sua requisição foi recebida com sucesso. Segue abaixo, o mapeamento dos possíveis “HTTP Code” retornados e seus significados.

Código	Mensagem
200	Informação processada com sucesso
400	Requisição com parâmetros obrigatórios vazios ou inválidos
401	Falha na autenticação ou sem acesso para usar o serviço
405	Método não permitido, o serviço suporta apenas POST
415	<i>Content-Type</i> não suportado
500	Erro fatal na aplicação, executar a solicitação mais tarde
503	Serviço está indisponível

**Tabela 3 - Códigos HTTP retornados para o cliente.**



## 5.2. Retorno em caso de sucesso

### 5.2.1. Informações retornadas

Nome	Tamanho	Tipo	Descrição e exemplo
Transaction	X	Objeto Raiz Serializado String	<p>Objeto Principal de retorno contendo todas as informações de resultado da requisição. Obs.: o objeto pode ser retornado serializado no formato XML ou JSON dependendo da escolha no momento da requisição (URL).</p> <p>Exemplo:</p> <pre>&lt;? xml version="1.0" encoding="UTF-8" ?&gt; &lt;transaction&gt;   &lt;transactionId&gt;90011430&lt;/transactionId&gt;   &lt;orderId&gt;123546&lt;/orderId&gt;   &lt;status&gt;1&lt;/status&gt;   &lt;descriptionStatus&gt;Em andamento&lt;/descriptionStatus&gt;   &lt;paymentLink&gt;https://www.bcash.com.br/checkout/Transferencia/Pagar/90011430/0P5he82M44tk7SWulKoa9j4o4hBpfX&lt;/paymentLink&gt; &lt;/transaction&gt;</pre>
transactionId	19	String	<p>Identificador da transação no Bcash.=</p> <p>Exemplo: &lt;transactionId&gt;90011430&lt;/transactionId&gt;</p>
orderId	50	String	<p>Identificador do número do pedido enviado pela loja ou inferido pelo Bcash (caso não seja informado, o Bcash considera esse valor como sendo o próprio identificador da transação).</p> <p>Exemplo: &lt;orderId&gt;123546&lt;/orderId&gt;</p>
Status	11	String	<p>Código do status da transação. Os valores podem ser:</p> <ul style="list-style-type: none"> <li>1 – Em andamento</li> <li>2 – Em andamento</li> <li>3 – Aprovada</li> <li>4 – Concluído</li> <li>5 – Cancelada</li> </ul> <p>Exemplo: &lt;status&gt;1&lt;/status&gt;</p>
descriptionStatus	255	String	<p>Descrição do status da transação. Os valores podem ser:</p>

			<p>1 – “Em andamento” = Aguardando a confirmação de pagamento</p> <p>2 – “Em andamento” = Aguardando aprovação de risco</p> <p>3 – “Aprovada” = Transação aprovada</p> <p>4 – “Concluído” = Transação concluída</p> <p>5 – “Cancelada” = Transação cancelada</p> <p>Exemplo:</p> <pre>&lt;descriptionStatus&gt;Em andamento&lt;/descriptionStatus&gt;</pre>
paymentLink	255	String	<p>Link para pagamento.</p> <p>Exemplo:</p> <pre>&lt;paymentLink&gt;https://www.bcash.com.br/checkout/T ransferencia/Pagar/90011430/0P5he82M44tk7SWul Koa9j4o4hBpfX&lt;/paymentLink&gt;</pre>

**Tabela 4 – Parâmetros de retorno em caso de sucesso**

## 5.2.2. Exemplos de retorno

XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<transaction>
  <transactionId>90012349</transactionId>
  <orderId>123546</orderId>
  <status>1</status>
  <descriptionStatus>Em andamento</descriptionStatus>

  <paymentLink>https://www.bcash.com.br/checkout/Ttransferencia/Pagar/90012349/0vvbjSO65XE6At5c9BY
  OGeEavzfM8</paymentLink>
</transaction>
```

JSON:

```
{"transactionId":"90012351","orderId":"123546","status":"1","descriptionStatus":"Em+andamento","paymentLink":"https%3A%2F%2Fwww.bcash.com.br%2Fcheckout%2FTtransferencia%2FPagar%2F90012351%2F0wTYTouKrkCUD6pFKH0hNDWoHyqgl7"}
```

Para o caso do formato JSON, as informações são transformadas através de encode HTML, sendo necessária a aplicação do decode para obter o valor original.

## 5.3. Retorno em caso de erro

### 5.3.1. Informações retorno de erro

Código	Mensagem
Código	Código do erro
Descrição	Descrição do erro

Tabela 5 – Parâmetros de retorno em caso de erro

## 5.4. Mensagens de Erro

As mensagens de erro são enviadas no processamento da solicitação do cliente, ou seja, quando o “HTTP Code” retornado for diferente de 200. Estas mensagens descrevem o código do erro e sua mensagem informativa e são enviadas no formato e codificação solicitados pelo cliente.

Código	Mensagem
102001	E-mail loja inválido
102002	E-mail inválido
102003	CPF informado é inválido
102004	CNPJ informado é inválido
102005	Telefone fixo inválido
102006	Celular inválido
102007	Estado inválido
102008	CEP inválido
102009	Sexo inválido
102010	Data Nascimento inválida

102011	RG inválido
102012	Data da emissão do RG inválida
102013	Estado emissor do RG inválido
102014	Token inválido
102015	Identificador da conta não é um número válido
102016	Nome inválido
102017	Endereço inválido
102018	Número do endereço inválido
102019	Cidade inválida
102020	Nome da empresa inválido
103001	Forma de Pagamento inválida
103002	Valor da transação é inválido
103003	Valor de moeda inválido
103004	Valor do frete inválido
103005	Valor do desconto inválido
103006	Valor do acréscimo inválido
103007	Valor do produto {0} é inválido
103008	Quantidade do produto {0} é inválida
103009	Número de parcelas inválido
103010	Número do cartão de crédito inválido
103011	Mês de vencimento do cartão de crédito está inválido
103012	Ano de vencimento do cartão de crédito está inválido
103013	Vencimento do cartão esta inválido

103014	Código de segurança do cartão de crédito está inválido
103015	Código da campanha inválido
103016	URL de retorno inválida
103017	URL de aviso inválida
103018	Valor de aceitação e/ou aceitação do contrato é inválido
103019	A transação referenciada pela transação dependente é inválida
103020	O id do pedido é obrigatório e deve ser maior que 0 e menor que 99999999
103021	O valor do pedido é obrigatório e deve ser maior que 0 e menor que 99999999.99
103022	O registrationNumber deve possuir 11 para CPF ou 14 posições para CNPJ
103023	O cep deve ser um valor numérico
103024	O cep deve ser um valor numérico de 8 posições
103025	Modalidade da transação cadastrada para o vendedor não suportada
202001	E-mail da loja é obrigatório
202002	Loja não encontrada
202003	Loja não pode ser o comprador
202004	Nome obrigatório
202005	E-mail obrigatório
202006	Sexo obrigatório
202007	CPF obrigatório
202008	CNPJ obrigatório
202009	Necessário informar ao menos um telefone
202010	Razão Social obrigatória

202011	Necessário enviar pelo menos um endereço
202012	Endereço obrigatório
202013	Número do endereço obrigatório
202014	Bairro obrigatório
202015	Cidade obrigatória
202016	Estado obrigatório
202017	CEP obrigatório
202018	Autenticação Basic obrigatória
202019	Falha na autenticação
202020	Necessário aceitar o contrato de prestação de serviço
202021	Cliente já cadastrado com outro cpf.
202022	Não foi encontrada nenhuma conta com o CPF e Email informado
202023	A conta do tipo: {0} está desativada
202024	O identificador da conta é obrigatório
202025	Token obrigatório
202026	ConsumerKey é obrigatório
202027	Autenticação é obrigatória
203001	Produto é obrigatório
203002	Valor do produto {0} é obrigatório
203003	Quantidade do produto {0} é obrigatória
203004	Código do produto {0} é obrigatório
203005	Descrição do produto {0} é obrigatória

203006	Forma de Pagamento obrigatório
203007	Meio de pagamento da tabela de tarifa inativo
203008	O meio de pagamento selecionado não está ativo para este cliente
203009	Número de parcelas é obrigatório
203010	Nome do titular do cartão é obrigatório
203011	Número do cartão de crédito é obrigatório
203012	Mês de vencimento do cartão de crédito é obrigatório
203013	Ano de vencimento do cartão de crédito é obrigatório
203014	Código de segurança do cartão de crédito é obrigatório
203015	Necessário informar os dados do cartão de crédito.
203016	Transação dependente com porcentagem incorreta.
101001	Versão da API é inválida ou está depreciada
101002	Parâmetro [data] deve seguir o padrão de formatação JSON
403001	Meio de pagamento indisponível no momento.
300001	Campo marcado como obrigatório está nulo {0}
302001	Objeto [Account] deve ser informado
302002	Objeto [Person] deve ser informado
302003	Objeto [Buyer] deve ser informado
302004	Objeto [Trade] deve ser informado
302005	Objeto [PersonJuridical] deve ser informado
302006	Objeto [Address] deve ser informado
303001	Objeto [Transaction] deve ser informado

303002	Objeto [SourceTransaction] deve ser informado
303003	Objeto [TransactionPurchase] deve ser informado
303004	Objeto [TransactionPurchaseData] deve ser informado
303005	Objeto [TransactionCreditCard] deve ser informado
303006	Objeto [CreditCard] deve ser informado
303007	Objeto [CreditCardCvv] deve ser informado
303008	Objeto [PaymentMethod] deve ser informado
301001	Parâmetros obrigatórios não preenchidos: {0}
301002	Parâmetro [data] é obrigatório
502001	Não foi possível criar a conta
502002	Não foi possível gerar a chave de acesso
502003	Não foi possível realizar a busca de contas por cpf
502004	Não foi possível atualizar a conta
503001	Não foi possível criar a transação
503002	Não foi possível calcular os valores da transação
503003	Não foi possível persistir os dados de boleto
503004	Não foi possível processar o pagamento
500000	Ocorreu um erro inesperado no sistema, por favor, tente mais tarde
500001	Serviço indisponível
602001	Chave de identificação não encontrada
602002	Acesso negado
602003	Falha na autenticação



602004	Assinatura de autenticação inválida
602005	Requisição duplicada [{0}]
602006	Requisição atrasada [{0}]
602007	Tipo de retorno inválido

**Tabela 6 – Códigos de mensagens de erro**

## 5.5 Exemplos de retorno

Exemplos de retorno em XML ou JSON, em caso de erro:

XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<set>
  <error>
    <code>202001</code>
    <description>E-mail da loja é obrigatório</description>
  </error>
</set>
```

JSON:

```
{"list":[{"code":"202001","description":"E-mail+da+loja+%C3%A9+obrigat%C3%B3rio"}]}
```

## 6. Exemplo de codificação

### 6.1 Códigos PHP para o serviço de criação de transação:

```
<?php
$time = time()*1000;
$microtime = microtime();
$rand = mt_rand();

$signature = array(
    "oauth_consumer_key"=>### Coloque aqui sua consumer key ###,
    "oauth_nonce"=>md5( $microtime . $rand ),
    "oauth_signature_method"=>"PLAINTEXT",
    "oauth_timestamp"=>$time,
    "oauth_version"=>"1.0",
);

$signature = base64_encode(http_build_query($signature, '', '&'));

$oauth =
array("Authorization: OAuth
    realm=https://api.bcash.com.br/service/createTransaction/xml/.
    ", "oauth_consumer_key=" . ### Coloque aqui sua consumer key ###.
    ", "oauth_nonce=" . md5( $microtime . $rand ).
    ", "oauth_signature=" . $signature.
    ", "oauth_signature_method=PLAINTEXT".
    ", "oauth_timestamp=" . $time.
    ", "oauth_version=1.0",
    "Content-Type:application/x-www-form-urlencoded; charset=" . ### Coloque aqui o seu charset ###,
);
$urlPost =
"https://api.bcash.com.br/service/createTransaction/xml/";
$data = ### Coloque aqui o objeto JSON com o TransactionRequest do pedido a #ser criado ex.:
{"products":[{"code":"1236456","description":"Produto de teste
1","amount":"5","value":"15.25","extraDescription":"Este produto é um
exemplo"}], "dependentTransactions":[{"email":"dependent@email.com", "value":"0.25"}], "buyer":{"address":{"
"address":"Rua teste teste", "number":"123", "complement":"apto. 000", "neighborhood":"Bairro de
exemplo", "city":"Exemplo de cidade", "state":"MG", "zipCode":"12345-
789"}, "mail":"comprador@comprador.com", "name":"Comprador Teste", "cpf":"850-822-365-04", "phone":"34-
3311-9999", "cellPhone":"34-9999-
1111", "gender":"M", "birthDate":"01/01/1970", "rg":"11337733", "issueRgDate":"01/01/1990", "organConsignor
Rg":"SSP", "stateConsignorRg":"MG", "companyName":"Empresa de teste", "cnpj":"72-139-715/0001-
30", "searchToken":"","paymentMethod":{"code":"58"}, "creditCard":null, "sellerMail":"lojamodelo@bcash.co
m.br", "orderId":"123546", "free":"Abc
def", "freight":"10.50", "freightType":"SEDEX", "discount":"1.95", "addition":"2.65", "urlReturn":"retorno.bcash.c
om.br", "urlNotification":"aviso.bcash.com.br", "installments":"1", "campaignId":"","currency":"BRL", "accepted
Contract":"S", "viewedContract":"S"}
###;
$version = "### Coloque aqui a versão da API a ser utilizada ex.: 1.0 ###";
```

```
$encode = ### Coloque aqui a codificação desejada (UTF-8 ou ISO-8859-1) ###;

$params = ### Coloque aqui os dados a serem enviados no formulário da requisição ex.:
array("data"=>$data,"version"=>$version,"encode"=>$encode) ###;

ob_start();
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $urlPost);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch,CURLOPT_POSTFIELDS, http_build_query($params, '', '&'));
curl_setopt($ch, CURLOPT_HTTPHEADER, $oAuth);
curl_exec($ch);

/* XML ou Json de retorno */
$resposta = ob_get_contents();

ob_end_clean();

/* Capturando o http code para tratamento dos erros na requisição*/
$httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if($httpCode != "200"){
//Tratamento das mensagens de erro
}else{
//Tratamento dos dados de resposta da consulta.
}
?>
```

## 6.2 Código JAVA para consumir o serviço de criação da transação:

Utilizando a LIB Java Bcash (Acrescentar as bibliotecas - full-pd-api-connector-[versão de sua escolha].jar e slf4j-api-1.6.4.jar - no classpath da aplicação).

```
import br.com.buscaped.connector.service.transaction.TransactionServiceImpl;
import br.com.buscaped.connector.service.TransactionService;
import br.com.buscaped.connector.domain.transaction.create.TransactionRequest;
import br.com.buscaped.connector.domain.transaction.create.TransactionResponse;
import br.com.buscaped.connector.domain.transaction.create.OAuthRequestData;

TransactionService transactionService = new TransactionServiceImpl();
// generateTransactionRequest método que gera um objeto do tipo TransactionRequest com
// as informações da transação, apenas com os setters das informações do objeto
// ex.:
// TransactionRequest transactionRequest = new TransactionRequest();
// transactionRequest.setSellerMail( SELLER_MAIL );
// transactionRequest.setOrderId( "123456" );
// transactionRequest.setFree( "Texto livre" );
// transactionRequest.setFreight( new BigDecimal( String.valueOf( random.nextInt( 99 )
// ) ) );
// transactionRequest.setFreightType( FREIGHT_TYPE ); e assim por diante . . .

TransactionRequest transactionRequest = generateTransactionRequest();
// CONSUMER_KEY = chave da sua loja
// com o objeto transactionResponse você acessa todas as informações de retorno da
// requisição
TransactionResponse transactionResponse = transactionService.createTransaction(
    transactionRequest,
    new OAuthRequestData( CONSUMER_KEY ) );
```