

GET/products/{productId} – Erro ExemploExpand

```
{  "transactionId": "abc123456",  "summary": {    "statusCode": 500,    "errors": [      {        "message": "Internal Server Error",        "type": "INTERNAL_SERVER_ERROR"      }    ]  }}
```

GET/skus/{skuId} – ExemploExpand

```
{  "transactionId": "abc123456",  "summary": {    "statusCode": 200  },  "sku": {    "id": 24229,    "name": "Mouse Microsoft Wedge Touch Mouse Bluetooth, Preto - 3LR-00013",    "active": true,    "eans": [      "12345ab", "67890ab"    ],    "specializations": [],    "buybox": {      "seller": {        "id": "1",        "name": "Walmart"      },      "active": true,      "available": true,      "listPrice": {        "BRL": 279      },      "discountPrice": {        "BRL": 188.1      }    },    "offers": [      {        "seller": {          "id": "1",          "name": "Walmart"        },        "active": true,        "available": true,        "listPrice": {          "BRL": 279        },        "discountPrice": {          "BRL": 188.1        }      },      {        "seller": {          "id": "teste",          "name": "teste"        },        "active": true,        "available": true,        "listPrice": {          "BRL": 279        },        "discountPrice": {          "BRL": 188.1        }      }    ]  },  "images": [    {      "url": "http://static.wmobjects.com.br/imgres/arquivos/ids/2490627",      "width": 1000,      "height": 1000    },    {      "url": "http://static.wmobjects.com.br/imgres/arquivos/ids/2490628",      "width": 1000,      "height": 1000    },    {      "url": "http://static.wmobjects.com.br/imgres/arquivos/ids/2490629",      "width": 1000,      "height": 1000    },    {      "url": "http://static.wmobjects.com.br/imgres/arquivos/ids/2490630",      "width": 1000,      "height": 1000    },    {      "url": "http://static.wmobjects.com.br/imgres/arquivos/ids/2490631",      "width": 1000,      "height": 1000    },    {      "url": "http://static.wmobjects.com.br/imgres/arquivos/ids/2490632",      "width": 1000,      "height": 1000    },    {      "url": "http://static.wmobjects.com.br/imgres/arquivos/ids/2490633",      "width": 1000,      "height": 1000    },    {      "url": "http://static.wmobjects.com.br/imgres/arquivos/ids/2490634",      "width": 1000,      "height": 1000    }  ],  "dimensions": {    "height": 1,    "length": 5,    "weight": 280,    "width": 14  }  }
```

GET/skus/{skuId} – Erro ExemploExpand

```
{  "transactionId": "abc123456",  "summary": {    "statusCode": 500,    "errors": [      {        "message": "Internal Server Error",        "type": "INTERNAL_SERVER_ERROR"      }    ]  }}
```

Simulate

Tem como objetivo permitir realizar simulações de compra, retornando informações atualizadas de preço, estoque, desconto e entrega para um conjunto de itens. Esse serviço é o que retorna garantidamente as informações mais recentes, e portanto recomendamos sua utilização sempre que se adicionar um item de novo caso dentro do "cartinho", por exemplo. Recomendamos sua utilização também previamente a operação de fechamento do pedido para garantir que os dados de preço, estoque, descontos e entregas estão consistentes, visando evitar falhas na operação.

Métodos Suportados

Recurso	Operação	Descrição
/simulate	POST	Serviço para realização de simulações de compra, retornando informações de preço, estoque, desconto e entrega para um conjunto de itens.

Considerações: Request

- Dados de entrega ("delivery") e de pagamento ("payment"), não são obrigatórios, porém devem ser passados previamente ao fechamento do pedido para evitar divergência de valores.
- Para clientes da modalidade Fidelidade Incentivo só será aceito o meio de pagamento pós pago "POST_PAID".
- Não é permitido fazer simulações ou efetuar pedidos com mais de 10 quantidades de um mesmo SKU. Essa regra é válida para todas as modalidades de venda dentro do Walmart. Caso seja informado mais de 10 unidades ("quantity"), é retornado erro 400 - Maximum 10 for each item quantity allowed.

Considerações: Response

- Caso o item em questão possua menos que 10 unidades em estoque, a quantidade retornada no atributo "stock" é a real, e o atributo "realStock" terá valor "true". Caso haja 10 ou mais unidades em estoque, a quantidade do atributo "stock" será sempre retornada com valor "10", e o atributo "realStock" com valor "false".
- A quantidade dos itens é utilizado para calcular o valor do frete do item.
- O preço retornado para cada oferta na simulação é equivalente ao valor unitário do item.
- O preço do frete ("delivery.slas[].price") já está calculado para a quantidade total do item.
- Caso um item esteja indisponível (sem estoque), o response retornará um status code 400, e um objeto de erro estará tanto na lista de erros do summary como na lista de erros do item. Nesses casos, os discounts não serão retornados para nenhum dos itens.

Exemplo erro 1Expand

```
{  "summary": {    "statusCode": 400,    "errors": [      {        "type": "STOCK_UNAVAILABLE",        "message": "Stock unavailable"      }    ]  },  "transactionId": "teste",  "itemsGroup": {    "items": [      {        "offer": {          "sellerId": "1",          "skuId": 1264020,          "availability": {            "error": {              "type": "STOCK_UNAVAILABLE",              "message": "Stock unavailable"            },            "stock": 0,            "realStock": true,            "price": {              "BRL": 1299            },            "listPrice": {              "BRL": 1499            }          }        },        "quantity": 1,        "discountPrice": {          "BRL": 1299        }      },      {        "discounts": []      }    ]  }  }
```

Exemplo erro 2Expand

```
{  "summary": {    "statusCode": 400,    "errors": [      {        "type": "LGTO115",        "message": "LGTO115: The item 2000013191 does not have stock"      }    ]  },  "transactionId": "teste",  "itemsGroup": {    "items": [      {        "offer": {          "sellerId": "1",          "skuId": 17537,          "availability": {            "stock": 2,            "realStock": true,            "price": {              "BRL": 629.88            },            "listPrice": {              "BRL": 999            }          }        },        "quantity": 1,        "discountPrice": {          "BRL": 629.88        },        "delivery": {          "error": {            "type": "LGTO115",            "message": "LGTO115: The item 2000013191 does not have stock"          },          "available": false        }      },      {        "discounts": []      }    ]  }  }
```

Obs.: Nesse segundo caso de erro foi realizado a simulação de um Kit, pode acontecer que um dos itens que compõem o Kit esteja sem estoque, para essa situação o correto é desconectar o valor do "stock" retornado no response.

- Caso ocorra um erro para a entrega do item (entrega não disponível para o endereço), o response retornará um status code 400, e um objeto de erro estará tanto na lista de erros do summary como na lista de erros do item. Nesses casos, os discounts não serão retornados para nenhum dos itens.

Exemplo de erroExpand

```
{  "summary": {    "statusCode": 400,    "errors": [      {        "type": "LGTO402",        "message": "LGTO402: Route not found for this address."      }    ]  },  "transactionId": "teste",  "itemsGroup": {    "items": [      {        "offer": {          "sellerId": "1",          "skuId": 1390321,          "availability": {            "stock": 10,            "realStock": false,            "price": {              "BRL": 1489            },            "listPrice": {              "BRL": 1599            }          }        },        "quantity": 1,        "discountPrice": {          "BRL": 1489        },        "delivery": {          "error": {            "type": "LGTO402",            "message": "LGTO402: Route not found for this address."          },          "available": false        }      },      {        "discounts": []      }    ]  }  }
```

Quando na simulação for informado uma quantidade de produto maior que o disponível no estoque, por exemplo, simulou com 9 itens mas só tem 4 no estoque, retornará o seguinte erro:

Exemplo de erroExpand

```
{  "summary": {    "statusCode": 400,    "errors": [      {        "type": "STOCK_OVERFLOW",        "message": "Stock overflow"      }    ]  },  "transactionId": "teste",  "itemsGroup": {    "items": [      {        "offer": {          "sellerId": "1",          "skuId": 22244,          "availability": {            "error": {              "type": "STOCK_OVERFLOW",              "message": "Stock overflow"            },            "stock": 9,            "realStock": true,            "price": {              "BRL": 60.75            },            "listPrice": {              "BRL": 68            }          }        },        "quantity": 10,        "discountPrice": {          "BRL": 60.75        }      },      {        "discounts": []      }    ]  }  }
```

Caso o produto não esteja disponível para a venda, retornará o seguinte erro:

Exemplo de erroExpand

```
{  "summary": {    "statusCode": 400,    "errors": [      {        "type": "INACTIVE",        "message": "INACTIVE: seller or offer is inactive"      }    ]  },  "transactionId": "teste",  "itemsGroup": {    "items": [      {        "offer": {          "sellerId": "1",          "skuId": 127,          "availability": {            "error": {              "type": "INACTIVE",              "message": "INACTIVE: seller or offer is inactive"            },            "realStock": true,            "price": null,            "listPrice": null          }        },        "quantity": 10,        "discountPrice": null      },      {        "discounts": []      }    ]  }  }
```

Erros

Código	Mensagem	Descrição
LGTO115	The item (id) does not have stock	Erro ocorre quando não há estoque disponível para um item.
LGTO402	Route not found for this address.	Erro ocorre quando não há transportadora que atenda o destinatário.
STOCK_UNAVAILABLE		Quando o produto não tem estoque.
STOCK_OVERFLOW		Quando simula uma quantidade de item maior que a disponível no estoque.
INACTIVE		Produto não está disponível para a venda.

Request

POST/simulate – Request ExemploExpand

```
{  "transactionId": "5vfd9pjymdc7pupombas5j7ve",  "locale": {    "currency": "BRL"  },  "itemsGroup": {    "items": [      {        "offer": {          "sellerId": "vtex_loja_modelo",          "skuId": 46393        },        "quantity": 9      },      {        "offer": {          "sellerId": "1",          "skuId": 999        },        "quantity": 2      }    ]  },  "delivery": {    "address": {      "country": "BRA",      "postalCode": "13209335"    }  },  "payment": {    "methods": [      {        "type": "CREDIT_CARD",        "name": "MASTERCARD"      }    ]  }  }
```

POST/simulate – Request SchemaExpand

```
SimulationRequest { transactionId (string,null, optional),  locale (SimulationRequestLocale),  itemsGroup (SimulationRequestItemsGroup),  delivery (SimulationRequestDelivery,optional),  payment (SimulationRequestPayment, optional) }SimulationRequestLocale {currency (string):Currency Code, 3 letters currency (ISO 4217), eg.: BRL, [maxLength = 3]}SimulationRequestItemsGroup {items (Array[SimulationRequestItem])}SimulationRequestDelivery {address (SimulationRequestAddress,optional)}SimulationRequestPayment {methods(Array[SimulationRequestMethod],optional)}SimulationRequestItem {offer (SimulationRequestOffer),  quantity (integer) used for the related offer, is used to calculate the price of delivery [minValue = 1]}SimulationRequestAddress {country (string):Delivery Address Country, [maxLength = 50],  postalCode (string):Delivery Address CEP, accept only numbers, [maxLength = 8]}SimulationRequestMethod {type (string):Payment Method Type,  ["CREDIT_CARD", "POST_PAID"],  name (string):Payment Method Name, [maxLength = 50]}SimulationRequestOffer {sellerId (string):seller Id, [maxLength = 50],  skuId (integer):SKU marketplace}
```

Response

POST/simulate – Response ExemploExpand

```
{  "transactionId": "5vfd9pjymdc7pupombas5j7ve",  "summary": {    "statusCode": 200  },  "itemsGroup": {    "items": [      {        "offer": {          "sellerId": "vtex_loja_modelo",          "skuId": 24421,          "availability": {            "stock": 10,            "realStock": false,            "price": {              "BRL": 10.53            },            "listPrice": {              "BRL": 10.53            }          }        },        "quantity": 10,        "discountPrice": {          "BRL": 10.53        },        "delivery": {          "available": false,          "slas": [            {              "id": 1,              "type": "NORMAL",              "duration": {                "value": "5",                "unit": "BUSINESS_DAY"              },              "price": {                "BRL": 16.54              }            },            {              "id": 5,              "type": "AGENDADA",              "duration": {                "value": "25",                "unit": "BUSINESS_DAY",                "restriction": "2015-09-30T15:27:52-03:00"              },              "schedules": [                {                  "begin": "2015-09-30T07:00:00-03:00",                  "end": "2015-09-30T12:00:00-03:00",                  "price": {                    "BRL": 10.53                  }                }              ]            }          ]          }        },        "rule": {          "id": 24438,          "target": "ITEM",          "type": "NOMINAL",          "nominal": 6.00        },        "value": 6.00      }    ]  },  "discounts": [    {      "rule": {        "id": 1,        "target": "ITEM",        "type": "PERCENT",        "percentage": 5      },      "value": 5    },    {      "rule": {        "id": 24438,        "target": "ITEMS_GROUP",        "type": "NOMINAL",        "nominal": 7.50      },      "value": 7.50    },    {      "rule": {        "id": 1451,        "target": "FREIGHT",        "type": "NOMINAL"      }    }  ]  }
```



```

"transactionId": "5vfd9pjymdc7pupombas5j7ve",
"summary": {
  "statusCode": 400,
  "errors": [
    {
      "type": "STOCK_OVERFLOW",
      "message": "Stock Overflow"
    },
    {
      "type": "DISCOUNT_GENERIC_ERROR",
      "message": "An error occurred in the discount calculation"
    }
  ]
},
"itemsGroup": {
  "items": [
    {
      "offer": {
        "sellerId": "vtex_loja_modelo",
        "skuid": 24421,
        "availability": {
          "error": {
            "code": "STOCK_OVERFLOW",
            "message": "STOCK_OVERFLOW"
          }
        },
        "stock": 10,
        "realStock": false,
        "price": {
          "BRL": 10.53
        },
        "listPrice": {
          "BRL": 10.53
        }
      }
    },
    {
      "quantity": 10,
      "discountPrice": {
        "BRL": 10.53
      },
      "delivery": {
        "available": false,
        "slas": [
          {
            "id": 1,
            "type": "NORMAL",
            "duration": {
              "value": 5,
              "unit": "BUSINESS_DAY"
            },
            "price": {
              "BRL": 6.54
            }
          },
          {
            "id": 5,
            "type": "AGENDADA",
            "duration": {
              "value": 25,
              "unit": "BUSINESS_DAY",
              "restriction": "2015-09-30T15:27:52-03:00"
            },
            "schedules": [
              {
                "begin": "2015-09-30T07:00:00-03:00",
                "end": "2015-09-30T12:00:00-03:00",
                "price": {
                  "BRL": 10.53
                }
              }
            ]
          }
        ]
      },
      "discounts": [
        {
          "rule": {
            "id": 124438,
            "target": "ITEM",
            "type": "NOMINAL",
            "nominal": 6.00
          },
          "value": 6.00
        }
      ]
    },
    {
      "discounts": [
        {
          "rule": {
            "id": 1,
            "target": "ITEM",
            "type": "PERCENT",
            "percentage": 5
          },
          "value": 5
        },
        {
          "error": {
            "type": "DISCOUNT_GENERIC_ERROR",
            "message": "An error occurred in the discount calculation"
          },
          "rule": {
            "id": 124438,
            "target": "ITEMS_GROUP",
            "type": "NOMINAL",
            "nominal": 7.50
          },
          "value": 7.50
        },
        {
          "rule": {
            "id": 1451,
            "target": "FREIGHT",
            "type": "NOMINAL"
          }
        }
      ]
    }
  ]
}
}

```

```
{
  "message": "Internal Server Error",
  "type": "INTERNAL_SERVER_ERROR"
}
}
```

```

type (string, optional)
)
SimulationResponseItem (offer (SimulationResponseOffer),
quantity (integer),
services (SimulationResponseServices, optional),
discountPrice (object):Price with discount applied, this value is equivalent to
the unit value of the item
,
delivery (SimulationResponseDelivery):Delivery information
discounts (Array[SimulationResponseDiscounts]):Discount item information
)
SimulationResponseDiscounts (error (SimulationResponseItemError, optional):return
if discount simulation failed
,
rule (SimulationResponseRule, optional):return if discount simulation were
successful and has applicable discount to this item
,
value (number):Nominal value (calculated) to be applied to each unit in item
)
SimulationResponseOffer (sellerId (string):seller Id
,
skuId (integer):SKU marketplace
,
availability (SimulationResponseAvailability):Offer Availability Information
)
SimulationResponseServices (id (integer),
quantity (integer),
availability (SimulationResponseAvailability),
price (object, optional):return if availability simulation were successful, maps
3 letters currency ISO 4217 to long value, eg.: BRL -> 1099
)
SimulationResponseDelivery (error (SimulationResponseItemError, optional):return
if delivery simulation failed
,
available (boolean):It indicates whether there is delivery availability
(including checking if there is any available form of delivery)
,
slas (Array[SimulationResponseSlas], optional):Delivery types availables, return
if delivery simulation were successful
)
SimulationResponseItemError (code (string, optional),
message (string, optional)
)
SimulationResponseRule {id (integer):Rule discount id
,
target (string):Where the discount is applied, possible values(enum{'ITEM',
'ITEMS_GROUP', 'FREIGHT'})
,
type (string, optional):Type of discount, possible values(enum {"PERCENT",
"MINIMAL", "FIXED"})
,
percentage (number, optional):only if is percent type
,
nominal (number, optional):only if is nominal type
,
fixedPrice (number, optional):only if is fixed type
}
SimulationResponseAvailability (error (SimulationResponseItemError,
optional):return if availability simulation failed
,
stock (integer, optional):Stock quantity
,
realStock (boolean, optional):Describe if the value of stock is real
,
price (object, optional):Unit price of offer (without discount)
,
listPrice (object, optional):Unit price before offer
)
SimulationResponseSlas {id (string): ,
type (string),
duration (SimulationResponseDuration),
price (object, optional):only if the delivery type is NOT scheduled, maps 3
letters currency ISO 4217 to long value, eg.: BRL -> 1099
,
schedules (Array[SimulationResponseSchedules], optional):only if is scheduled
delivery type
}
SimulationResponseDuration (value (string):Estimated delivery value. Number or
date ISO 8601 (pattern: 'yyyy-MM-dd'T'HH:mm:ssXXX'), ex: '1',
'2015-06-11T15:27:52-03:00'
,
unit (string):Estimated delivery unit. Enum {DAY, BUSINESS_DAY, WEEK, HOUR, DATE}
,
restriction (string):Deadline for delivery type. Date ISO 8601 (pattern:
'yyyy-MM-dd'T'HH:mm:ssXXX'), ex: '2015-06-11T15:27:52-03:00'
)
SimulationResponseSchedules (begin (string):Initial date. Date ISO 8601 (pattern:
'yyyy-MM-dd'T'HH:mm:ssXXX')
,
end (string):End date. Date ISO 8601 (pattern: 'yyyy-MM-dd'T'HH:mm:ssXXX')
,
price (object):Schedule price. Maps 3 letters currency ISO 4217 to long value,
eg.: BRL -> 1099
)
}

```

payments

Retorna lista de meios de pagamento que podem ser utilizados para uma compra

Recurso	Operação	Descrição
paymentMethods	POST	A partir de uma lista de itens que compõe uma compra, retorna lista de meios de pagamento que podem ser utilizados

considerações: Response

- Para clientes da modalidade Fidelidade Incentivo só é disponível o meio de pagamento pós pago "POST_PAID".

request

```

POST: /paymentMethods — Request Exemplo
> Expand

```

1

```

        "sellerId": "vtex_loja_modelo",
        "skuId": 46393
    },
    "quantity": 20
},
{
    "offer": {
        "sellerId": "1",
        "skuId": 999
    }
}

```

$$\left. \begin{array}{l} \text{ } \\ \text{ } \end{array} \right\} \text{quant}$$

```

    }
}

POST: /paymentMethods — Request Schema Expand
SimulationRequestItemsGroup {
  items: Array<SimulationRequestItem> source
}

```

	offer
	quant
	calculate

- ```

 }
 SimulationRequestOffer {
 sellerId(string): seller Id, [maxLength = 50],
 skuId(integer): SKU marketplace
 }
}

```

Respon

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |  |        |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--------|
|  | POST: /paymentMethods — Response Exemplo                                                                                                                                                                                                                                                                                                                                                                                                                    |  | Expand |
|  | <pre> {   "summary": {     {       "statusCode": 200     },     "paymentMethods": {       {         "methods": {           {             {               "name": "VISA",               "type": "CREDIT_CARD"             },             {               "name": "MASTERCARD",               "type": "CREDIT_CARD"             }           }         }       }     }   } }</pre>                                                                             |  |        |
|  | POST: /paymentMethods — Response Schema                                                                                                                                                                                                                                                                                                                                                                                                                     |  | Expand |
|  | <pre> PaymentMethodsResponse {   transactionId(string, optional),   summary(Summary),   paymentMethods(PaymentMethods, optional) } Summary {   statusCode(integer),   errors(Array[SummaryErrors], optional):   Errors list } PaymentMethods {   methods(Array[PaymentMethodsMethod], optional) } SummaryErrors {   message(string, optional),   type(string, optional) } PaymentMethodsMethod {   name(string, optional),   type(string, optional) }</pre> |  |        |
|  | POST: /paymentMethods — Erro Schema                                                                                                                                                                                                                                                                                                                                                                                                                         |  | Expand |
|  | <pre> {   "summary": {     "statusCode": 400,     "errors": {       {         "type": "INVALID_PAYMENT_METHODS",         "message": "Maximum 1 payment method allowed"       }     }   } }</pre>                                                                                                                                                                                                                                                            |  |        |

- Os campos n
- As formas de
- que hajam fo

## Request

```
POST /installments - Request Example
{
 "itemsGroup": {
 "items": [
 {
 "id": "1",
 "name": "Item 1",
 "price": 100,
 "quantity": 1
 },
 {
 "id": "2",
 "name": "Item 2",
 "price": 200,
 "quantity": 1
 }
]
 }
}
```

```

 {
 "offer": {
 "sellerId":

```

```

 "skuId": 34789
 },
 "discountPrice": {
 "BRL": 5000
 },
 "quantity": 1
 }
],
 "payment": {

```

```

"methods": [
 {
 "type": "CR
 }
]

```

- ```

    "name": "HIFERCARD",
    "value": 5000,
    "bin": "606283"
  }
}
}
}

```
- ### POST /installments - Request Schema

[Expand](#)
- ```

InstallmentRequest {
 itemGroup (InstallmentRequestItemGroup),
 source
}

```

```

 payment (Installment
 }
 InstallmentRequestIt
 items (2max[Instal

```

- ```

    items (Array[InstallmentRequestItem])
  }
  InstallmentRequestPayment {
    methods (Array[InstallmentRequestPaymentMethod]): List of payments methods, {min
= 1, max = 2}
  }
  InstallmentRequestItem {

```

```
discountPrice (object)
quantity (number, nu
```

```

InstallmentRequestPaymentMethod {
  type (string): Payment Method Type = ['CREDIT_CARD'],
  name (string, optional): Payment Method Name, [maxLength = 50]. Is required if
  bin is filled,
  value (number): Value, [maxValue = 9999999999999999, Max decimal digits = 2],
  bin (string, optional): Credit Card bin, [size = 6]
}
InstallmentRequestOffer {
  sellerId (string): seller Id, [maxLength = 50],
  skuId (integer): SKU marketplace
}

```


POST /installments - Response Example

Expand

source

```
{
  "summary": {
    "statusCode": 200
  },
  "payments": [
    {
      "paymentType": {
        "name": "HIPERCARD",
        "type": "CREDIT_CARD",
        "maxCet": 27.58,
        "installments": [
          {
            "totalValue": {
              "BRL": 50
            },
            "valuePerInstallment": {
              "BRL": 50
            },
            "installmentAmount": 1
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 51.38
            },
            "valuePerInstallment": {
              "BRL": 25.69
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 1.38
            },
            "installmentAmount": 2
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 51.75
            },
            "valuePerInstallment": {
              "BRL": 17.25
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 1.75
            },
            "installmentAmount": 3
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 52.16
            },
            "valuePerInstallment": {
              "BRL": 13.04
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 2.16
            },
            "installmentAmount": 4
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 52.55
            },
            "valuePerInstallment": {
              "BRL": 10.51
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 2.55
            },
            "installmentAmount": 5
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 52.92
            },
            "valuePerInstallment": {
              "BRL": 8.82
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 2.92
            },
            "installmentAmount": 6
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 53.34
            },
            "valuePerInstallment": {
              "BRL": 7.62
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 3.34
            },
            "installmentAmount": 7
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 53.68
            },
            "valuePerInstallment": {
              "BRL": 6.71
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 3.68
            },
            "installmentAmount": 8
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 54.09
            },
            "valuePerInstallment": {
              "BRL": 6.01
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 4.09
            },
            "installmentAmount": 9
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 54.5
            },
            "valuePerInstallment": {
              "BRL": 5.45
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 4.5
            },
            "installmentAmount": 10
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 54.89
            },
            "valuePerInstallment": {
              "BRL": 4.99
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 4.89
            },
            "installmentAmount": 11
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 55.32
            },
            "valuePerInstallment": {
              "BRL": 4.61
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 5.32
            },
            "installmentAmount": 12
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 55.64
            },
            "valuePerInstallment": {
              "BRL": 4.28
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 5.64
            },
            "installmentAmount": 13
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 56.14
            },
            "valuePerInstallment": {
              "BRL": 4.01
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 6.14
            },
            "installmentAmount": 14
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 56.55
            },
            "valuePerInstallment": {
              "BRL": 3.77
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 6.55
            },
            "installmentAmount": 15
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 56.96
            },
            "valuePerInstallment": {
              "BRL": 3.56
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 6.96
            },
            "installmentAmount": 16
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 57.29
            },
            "valuePerInstallment": {
              "BRL": 3.37
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 7.29
            },
            "installmentAmount": 17
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 57.78
            },
            "valuePerInstallment": {
              "BRL": 3.21
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 7.78
            },
            "installmentAmount": 18
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 58.14
            },
            "valuePerInstallment": {
              "BRL": 3.06
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 8.14
            },
            "installmentAmount": 19
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 58.6
            },
            "valuePerInstallment": {
              "BRL": 2.93
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 8.6
            },
            "installmentAmount": 20
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 59.01
            },
            "valuePerInstallment": {
              "BRL": 2.81
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 9.01
            },
            "installmentAmount": 21
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 59.4
            },
            "valuePerInstallment": {
              "BRL": 2.7
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 9.4
            },
            "installmentAmount": 22
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 59.8
            },
            "valuePerInstallment": {
              "BRL": 2.6
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 9.8
            },
            "installmentAmount": 23
          },
          {
            "totalValue": {
              "BRL": 50
            },
            "totalValueWithRate": {
              "BRL": 60.24
            },
            "valuePerInstallment": {
              "BRL": 2.51
            },
            "rate": 1.5,
            "rateValue": {
              "BRL": 10.24
            },
            "installmentAmount": 24
          }
        ]
      }
    }
  ]
}
```

POST /installments - Response Error Example

Expand

source

```
{
  "summary": {
    "statusCode": 404,
    "errors": [
      {
        "type": "NOT_FOUND",
        "message": "Not Found"
      }
    ]
  }
}
```

POST /installments - Response Schema

Expand

source

```
InstallmentResponse {
  summary (Summary)
  payments (Array<InstallmentPaymentMethods>, optional)
  summary {
    statusCode (integer),
    errors (Array<SummaryErrors>, optional): Errors list
  }
  InstallmentPaymentMethods {
    paymentType (InstallmentPaymentMethodsType, optional): The
    maxCet (number, optional): The
    maximum effective total cost value, [maxValue = 9999999999999999, Max decimal
    digits = 2]. This field is required when installments with interest,
    installments (Array<InstallmentBody>, optional)
  }
  SummaryErrors {
    message (string, optional),
    type (string, optional)
  }
  InstallmentPaymentMethodsType {
    name (string, optional),
    type (string, optional)
  }
  InstallmentBody {
    totalValue (object, optional),
    totalValueWithRate (object, optional),
    valuePerInstallment (object, optional),
    rate (number, optional),
    rateValue (object, optional),
    installmentAmount (integer, optional)
  }
}
```

POST /installments - Response Error Example

Expand

source

```
{
  "summary": {
    "statusCode": 404,
    "errors": [
      {
        "type": "NOT_FOUND",
        "message": "Not Found"
      }
    ]
  }
}
```

POST /installments - Response Schema

Expand

source

```
InstallmentResponse {
  summary (Summary)
  payments (Array<InstallmentPaymentMethods>, optional)
  summary {
    statusCode (integer),
    errors (Array<SummaryErrors>, optional): Errors list
  }
  InstallmentPaymentMethods {
    paymentType (InstallmentPaymentMethodsType, optional): The
    maxCet (number, optional): The
    maximum effective total cost value, [maxValue = 9999999999999999, Max decimal
    digits = 2]. This field is required when installments with interest,
    installments (Array<InstallmentBody>, optional)
  }
  SummaryErrors {
    message (string, optional),
    type (string, optional)
  }
  InstallmentPaymentMethodsType {
    name (string, optional),
    type (string, optional)
  }
  InstallmentBody {
    totalValue (object, optional),
    totalValueWithRate (object, optional),
    valuePerInstallment (object, optional),
    rate (number, optional),
    rateValue (object, optional),
    installmentAmount (integer, optional)
  }
}
```

Visando atender requisitos de segurança no tráfego de informações de pagamento - segundo o padrão PCI, a API expõe um serviço para transformar os dados sensíveis de cartões de créditos em um token. Após o token ser gerado, ele é tratagado nas operações subsequentes relacionados ao processamento de um pedido - representando assim as informações do cartão de crédito. Essa operação é obrigatória dentro fluxo, para pedidos que envolvam pagamento com cartão de crédito, e visa proteger/soliar tais informações.

Métodos suportados

Recurso	Operação	Descrição
/tokenizer	POST	Serviço para geração do token do cartão de crédito.

Considerações: Request

- O serviço /tokenizer não está disponível para usuários da modalidade Fidelidade Incentivo (FI).

Considerações: Response

- O valor retornado deve ser utilizado na operação de inclusão de pedidos, no atributo correspondente.
- Os valores gerados tem validade de 5 minutos para serem utilizados nas operações de inclusão de pedidos.

Request

POST /tokenizer - Request Exemplo

Expand

source

```
{
  "creditCardNumber": "4989438843884305",
  "securityCode": "737"
}
```

POST /tokenizer - Request Schema

Expand

source

```
TokenizerRequest{
  creditCardNumber (string):Credit card number,
  [
    maxLength = 21,
    minLength = 13,
    only numbers
  ],
  securityCode (string):Credit card security code,
  [
    maxLength = 4,
    minLength = 3,
    only numbers
  ]
}
```

Response

POST /tokenizer - Response Exemplo

Expand

source

```
{
  "summary":{
    "statusCode":200
  },
  "encryptedNumber":"498943HTKLKE4305"
}
```

POST /tokenizer - Erro Exemplo

Expand

source

```
{
  "summary":{
    "statusCode":500,
    "errors":{
      {
        "message":"Internal Server Error",
        "type":"INTERNAL_SERVER_ERROR"
      }
    ]
  }
}
```

POST /tokenizer - Response Schema

Expand

source

```
TokenizerResponse {
  summary (Summary),
  encryptedNumber (string, optional): Credit card encrypted number
}
summary {
  statusCode (integer),
  errors (Array<SummaryErrors>, optional): Errors list
}
SummaryErrors {
  message (string, optional),
  type (string, optional)
}
```

Orders

Serviço que permite a criação e consulta de pedidos.

Métodos suportados

Recurso	Operação	Descrição
/orders	POST	Serviço para criação dos pedidos. Como entrada devem ser fornecidos dados do cliente da compra, dados de pagamento, e dos itens que fazem parte do pedido.
/orders/tracking/{id}	GET	Serviço para pesquisa de dados/acompanhamento de pedidos. São expostos dados referentes ao processamento do pagamento, entrega, dados pertinentes a nota fiscal do pedido e status do pedido.

Considerações

Considerações gerais:

- Recomenda-se a realização de uma operação de "simulação", previamente a criação do pedido - para garantir que os dados de preço, disponibilidade, entrega, continuam os mesmos. Tais dados são validados na operação de inclusão do pedido, e caso haja alguma inconsistência, uma mensagem de erro pertinente será retornada na operação
- Os dados de itens do pedidos devem ser incluídos de acordo com o retornado na API de "simulação", incluindo inclusive os descontos retornados.
- O atributo "genericOrderId" refere-se ao identificador do pedido para quem está invocando a API. Deve ser único para cada pedido gerado.
- O valor total do(s) pagamento(s) aplicado(s) a compra deve ser **igual** a soma de todos os itens (somados ao custo de entrega selecionado)
- O atributo "paymentMethod" deve ser preenchido com o valor correspondente do atributo "type" do meio de pagamento selecionado (acesse a seção "Payment API" desse documento para maiores informações). Caso o meio de pagamento selecionado for Cartão de crédito/CREDIT_CARD) o objeto creditCard deve estar preenchido.
- Não é permitido fazer simulações ou efetuar pedidos com mais de 10 quantidades de um mesmo SKU. Essa regra é válida para todas as modalidades de venda dentro do Walmart. Caso seja informado mais de 10 unidades ("quantity"), é retornado erro 400 - Maximum 10 for each item quantity allowed.
- Para os atributos de endereço ("postalCode", "city" e "state") atente-se para sempre enviar valores condizentes. Enviar um CEP que não coincide com a cidade/estado fornecido gera falha no processo de inclusão/processamento do pedido. Para os atributos "number" de endereço, fornecer somente valores numéricos (para demais informações como quadra, lote, etc., utilizar o atributo "complement").
- Para pedidos FI, sugerimos que os dados de "shipping" sejam os mesmos dados do "customer" para garantir maior fluidez na rastreabilidade do pedido.

Considerações para pedidos com cartão de crédito, deve-se utilizar o atributo "payments": os dados previamente obtidos a partir da API de parcelamento (instalments API). Visando atender requisitos de segurança no tráfego de informações de pagamento, **não devem ser fornecidos** na criação do pedido: número do cartão de crédito, e código de segurança do cartão. Deve-se primeiramente "tokenizar" tais dados na operação descrita em "Tokenizer API". O token gerado, deve ser fornecido no atributo: "encryptedNumber".

- Para pagamentos com cartão de crédito, os valores ligados a parcelamento, passados dentro do atributo "payments", **devem s**er repassados a partir dos códigos de API de parcelamento (Ex: "totalValue","totalValueWithRate","rateValue","rate","maxCet".)
- O campo **maxCet** é obrigatório quando o pedido tem juros, em pedidos sem juros o campo não deve ser informado.
- Atualmente para compras com cartão de crédito, são suportadas compras com até 2 cartões para um pedido.

Restrições:

- Atualmente a API aceita somente pedidos onde o consumidor seja pessoa física. Isso implica que o campo "document" do item "customer" aceita somente valores para CPF (e não para CNPJ)

Request

POST /orders - Pedido com Pagamento com cartão de crédito - Request Exemplo	Expand
source	
<pre>{ "transactionId":"asdasdasdasd", "genericOrderId":"12312312-xxx", "locale":{ "currency":"BRL" }, "customer":{ "firstName":"Jose chaves", "lastName":"Silva", "birthDate":"01/01/2001", "gender":"M", "type":"INDIVIDUAL", "email":"jose.chaves@gmail.com", "phones":[{ "type":"MOBILE", "number":"199999999" }], "document":"42715960433", "stateInscription":null }, "payments":[{ "totalValue":30.35, "valuePerInstallment":133.39, "rateValue":3.04, "rate":10.0, "paymentMethod":"CREDIT_CARD", "billingAddress":{ "address":"Rua Teste", "number":"12344", "complement":"Ap. 01, proximo a", "neighborhood":"bairro", "postalCode":"06335030", "city":"Caracas", "state":"SP", "country":"BRA", "type":"RESIDENTIAL" }, "creditCard":{ "installmentAmount":2, "valuePerInstallment":16.70, "maxCET":2.8, "encryptedNumber":"4xxxxxxxxxxxxx1", "holder":{ "name":"Teste" }, "brand":"VISA", "expirationMonth":"06", "expirationYear":"16" } }], "shipping":{ "recipient":"Maria da Silva", "recipientPhone":"55114115588", "recipientDocument":"34234342323", "recipientEmail":"mariasilva@gmail.com", "type":"CPF", "address":{ "address":"Rua", "number":"12344", "complement":"Ap. 01", "neighborhood":"bairro", "reference":"proximo a", "postalCode":"06335030", "city":"Caracas", "state":"SP", "country":"BRA", "type":"RESIDENTIAL" } }, "itemsGroup":{ "items":[{ "offer":{ "sellerId":"vtex_loja_modelo", "skuId":24421, "price":{ "BRL":10.53 } }, "quantity":1, "discountPrice":{ "BRL":4.53 }, "sla":{ "id":5, "type":"AGENDADA", "duration":{ "unit":"BUSINESS_DAY", "value":"18" }, "schedule":{ "begin":"2015-09-30T07:00:00-03:00", "end":"2015-09-30T12:00:00-03:00" }, "price":{ "BRL":3.20 } }, "discounts":[{ "rule":{ "id":24438, "target":"ITEM", "type":"NOMINAL", "nominal":6.00 }, "value":6.00 }], "offer":{ "sellerId":"I", "skuId":2879, "price":{ "BRL":20.49 } }, "quantity":1, "discountPrice":{ "BRL":120.49 }, "sla":{ "id":5, "type":"Normal", "duration":{ "unit":"BUSINESS_DAY", "value":"5" }, "restriction":"2015-09-30T10:30:00-03:00" }, "price":{ "BRL":2.13 } }], "discounts":[{ "rule":{ "id":24438, "target":"ITEMS_GROUP", "type":"NOMINAL", "nominal":7.50 }, "value":7.50 }, { "rule":{ "id":1451, "target":"FREIGHT", "type":"NOMINAL" } }] } }</pre>	

POST /orders - Pedido Pós Pago - Request Exemplo	Expand
source	
<pre>{ "transactionId": "012312339", "genericOrderId": "54545431434325", "locale": { "currency": "BRL" }, "customer": { "firstName": "Jose chaves", "lastName": "Silva", "birthDate": "01/01/2001", "gender": "M", "type": "INDIVIDUAL", "email": "jose.chaves@gmail.com", "phones": [{ "type": "MOBILE", "number": "1199999999" }], "document": "42715960433" }, "payments": [{ "totalValue": 149.9, "paymentMethod": "POST_PAID", "billingAddress":{ "address":"Rua Teste", "number":"12344", "complement":"Ap. 01, proximo a", "neighborhood":"bairro", "postalCode":"06335030", "city":"Caracas", "state":"SP", "country":"BRA", "type":"RESIDENTIAL" } }], "shipping": { "recipient": "Teste CVV 02", "recipientPhone": "19999898989", "recipientDocument": "07731514005", "recipientEmail": "teste@walmart.com.br", "type": "NORMAL", "address": { "address": "Rua Teste", "number": "123", "complement": "Jardim São Paulo(Zona Leste)", "neighborhood": "Jardim São Paulo(Zona Leste)", "reference": "Jardim São Paulo(Zona Leste)", "postalCode": "13183480", "city": "Hortolandia", "state": "SP", "country": "BRA", "type": "RESIDENTIAL" } }, "itemsGroup": { "items": [{ "offer": { "sellerId": "I", "skuId": 884418, "price": { "BRL": 149.9 } }, "quantity": 1, "discountPrice": { "BRL": 149.9 }, "discounts": [{ "sla": { "id": "I", "type": "Normal", "duration": { "unit": "BUSINESS_DAY", "value": "5" }, "price": { "BRL": 0 } }] } }] } }</pre>	

POST /orders - Request Schema	Expand
source	
<pre>OrderRequest { transactionId (string,null, optional): Transaction ID, [maxLength = 36] genericOrderId (string): Generic Order ID, [maxLength = 60] locale (OrderLocale): Locale info customer (OrderCustomer): Customer payments (Array[OrderPayment]): Payments [min=1, max=2] shipping (OrderShipping): Shipping itemsGroup (OrderItemsGroup): Items Group orderLocale (OrderLocale { currency (string): Currency, [maxLength = 3] } OrderCustomer { firstName (string): First Name, must not contains only numbers, [maxLength = 50] lastName (string): Last Name, must not contains only numbers, [maxLength = 50] type (string): Customer Type = ['INDIVIDUAL'], email (string): Email, examples = 'partner@walmart.com' [maxLength = 100] phones (Array[OrderPhone]): Phones, [minLength = 1] document (string): Customer Document (CPF), only numbers, [maxLength = 11] stateInscription (string,null, optional): State Inscription, [maxLength = 50] birthDate (string,null, optional): Birth Date, Date (pattern: 'dd/MM/yyyy', ex: '10/03/1992' gender (string,null, optional): Gender, [maxLength = 1] } OrderPayment { totalValue (number): Total Value [Max decimal digits = 2] totalValueWithRate (number,null, optional): Total Value with Rate [Max decimal digits = 2]. This field is required when installments with interest rate (number,null, optional): Rate [Max decimal digits = 2]. This field is required when installments with interest rateValue (number,null, optional): Rate Value [Max decimal digits = 2]. This field is required when installments with interest paymentMethod (string): Payment Method = ['CREDIT_CARD', 'POST_PAID'], billingAddress (OrderAddress, optional): Billing Address creditCard (OrderCreditCard, optional): Credit Card. This field is required when payment method is CREDIT_CARD } OrderShipping { recipient (string): Destinationary name, must not contains only numbers, [maxLength = 100] recipientTitle (string,null, optional): Destinationary title, [maxLength = 15] recipientPhone (string,null, optional): Destinationary phone, pattern = {+(optional) DDI(optional)}(optional) DDD number(8 or 9 digits), examples = '+5519998687363, 5519998687363 or 19998687363' [minlength = 10, maxLength = 14] recipientDocument (string,null, optional): Destinationary document, accept CPF or CNPJ, [minLength = 11, maxLength = 14] recipientEmail (string,null, optional): Destinationary email, examples = 'partner@walmart.com' [maxLength = 100] type (string,null, optional): Destinationary document type, [maxLength = 50] address (OrderAddress): Shipping Address } OrderItemsGroup { items (Array[OrderItem]): List of items, [min = 1] discounts (Array[OrderDiscount], optional): List of discounts } OrderPhone { number (string): Phone number, pattern = {+(optional) DDI(optional)}(optional) DDD number(8 or 9 digits), examples = '+5519998687363, 5519998687363 or 19998687363' [minLength = 10, maxLength = 14] type (string): Phone type = ['HOME', 'WORK', 'MOBILE'] } OrderAddress { address (string): Destinationary address, [maxLength = 150] number (string,null, optional): Destinationary address number, accept only numbers, [maxLength = 6] complement (string,null, optional): Destinationary address complement, [maxLength = 100] neighborhood (string): Destinationary address neighborhood, [maxLength = 50] reference (string,null, optional): Reference address, [maxLength = 100] postalCode (string): Destinationary address postal code, accept only numbers, [maxLength = 8] city (string): Destinationary address city, [maxLength = 100] state (string): Destinationary address state, [maxLength = 10] type (string): Destinationary address type = ['RESIDENTIAL', 'BUSINESS'], country (string): Destinationary address country, [maxLength = 5] } OrderCreditCard { installmentAmount (integer): Number of installments, [minValue = 1] valuePerInstallment (number): Value per Installment [Max decimal digits = 2] maxCET (number,null, optional): Max CET (Máximo Custo Efetivo) [Max decimal digits = 2]. This field is required when installments with interest encryptedNumber (string): Encrypted credit card number, [minLength = 6, maxLength = 200, only alphanumeric, underscore, dot and hyphen] brand (string): Credit card brand, [maxLength = 50] expirationMonth (string): Month expiration, only numbers, [minLength = 1], [maxLength = 2] expirationYear (string): Year expiration, only numbers, [minLength = 2], [maxLength = 4] holder (OrderCardHolder): Holder } OrderItem { offer (OrderOffer): Item Offer quantity (integer): Item quantity, [minValue = 1, maxValue = 10] discounts (Array[OrderDiscount], optional): List of discounts discountPrice (object): Item price with discounts, [minSize = 1] sla (OrderSla): Item Sla } OrderDiscount { rule (OrderDiscountRule): Discount rule value (number,null, optional): Discount value } OrderCardHolder { name (string): Holder name, must not contains only numbers, [maxLength = 50] document (string,null, optional): Holder Document, accept CPF or CNPJ, [minLength = 11, maxLength = 14] } OrderOffer { sellerId (string): Seller id, [maxLength = 50] skuId (integer): Sku id, [minValue = 1] price (object): Offer price, [minSize = 1] } OrderSla { id (string): Sla id, [maxLength = 100] type (string): Sla type, [maxLength = 100] duration (OrderSlaDuration): Sla duration price (object, optional): Sla price (required for no Scheduled SLAs) schedule (OrderSchedule, optional): Sla schedule } OrderDiscountRule { id (integer,null, optional): Discount rule id target (string): Discount target = ['ITEM', 'ITEMS_GROUP', 'FREIGHT'], type (string): Discount value type value = ['PERCENT', 'NOMINAL', 'FIXED'], percentage (number,null, optional): Discount percentage value. Required when type=PERCENT nominal (number,null, optional): Discount nominal value. Required when type=NOMINAL fixedPrice (number,null, optional): Discount fixed price value. Required when type=FIXED } OrderSlaDuration { unit (string): Sla duration unity = ['DAY', 'BUSINESS_DAY', 'WEEK', 'HOOR', 'DATE'] string Enum: "DAY", "BUSINESS_DAY", "WEEK", "HOOR", "DATE" value (string): Sla duration value. Number or date ISO 8601 (pattern: 'yyyy-MM-dd'T'HH:mm:ssXXX'), ex: '1', '2015-06-11T15:27:52-03:00' restriction (string,null, optional): Sla duration restriction. Date ISO 8601 (pattern: 'yyyy-MM-dd'T'HH:mm:ssXXX'), ex: '2015-06-11T15:27:52-03:00' } OrderSchedule { begin (string): Begin date, Date ISO 8601 (pattern: 'yyyy-MM-dd'T'HH:mm:ssXXX'), ex: '2015-06-11T15:27:52-03:00' end (string): End date, Date ISO 8601 (pattern: 'yyyy-MM-dd'T'HH:mm:ssXXX'), ex: '2015-06-11T15:27:52-03:00' price (object): Schedule price, [minSize = 1] } }</pre>	

Response

POST /orders - Response Exemplo	Expand
source	
<pre>{ "transactionId":"asdasdasdasd", "summary":{ "statusCode":200 }, "orderId": 268878329 }</pre>	
POST /orders - Erro Exemplo	Expand
source	
<pre>{ "transactionId": "abc123456", "summary": { "statusCode": 500, "errors": [{ "message": "Internal Server Error", "type": "INTERNAL_SERVER_ERROR" }] } }</pre>	
POST /orders - Erro Exemplo GenericOrderId duplicado	Expand
source	
<pre>{ "summary": { "errors": [{ "type": "OMS_PLACE_ORDER_ERROR", "message": "Duplicated genericOrderId" }], "statusCode": 400 }, "transactionId": "sp_16929", "orderId": 50099249 }</pre>	
POST /orders - Response Schema	Expand
source	
<pre>OrderResponse { transactionId (string, optional), summary (Summary), orderId (integer, optional) } Summary { statusCode (integer), errors (Array[SummaryErrors], optional): Errors list } SummaryErrors { message (string, optional), type (string, optional) }</pre>	

Tabela com principais códigos de erro retornados na operação:

Código	Descrição
OMS006	Conexão a dependência do fluxo com tempo excedido
PLC0201	Não existe quantidade disponível para esse item
PLC0213	O custo do frete mudou
PLC0219	Falha na operação de pré-autorização de compra com cartão de crédito junto a operadora
PLC0220	Falha na operação de pré-autorização de compra com cartão de crédito junto a operadora
PLC0721	Valor total fornecido no pedido não corresponde ao valor total dos itens, com seus respectivos encargos (frete, descontos, etc)
POV0001	Ocorreu erro com os dados de requisição enviados

CST_CUSTOMER_REGISTRATION_ERROR	Ocorreu falha no processamento das informações passadas no pedido, referentes ao cliente da compra (OrderCustomer). Cheque o que está descrito em "Restrições" nessa documentação para maiores informações.
---------------------------------	---

OMS_PLACE_ORDER_ERROR	Esse erro pode ser retornado com a mensagem "Duplicated genericOrderId", que representa uma falha na geração do pedido pois já existe um pedido criado com o ge nericOrderId informado.
-----------------------	---

Exemplo de Pedido identificando campos e origem dos mesmos:

Características:

- 2 Itens **SellerId 1 (WM): Smartphone LG Leon H326TV**

- **Pagamento com Cartão de Crédito (HIPERCARD)**

- **Parcelado em 18x**

- **C/ Desconto Percentual (PERCENT)**

- **Frete Normal e**

- **C/ Desconto de Carinho (NOMINAL)**

POST/orders - Response Exemplo

Expand

source

```
{
  "genericOrderId": "Exemplo_Pedido_com_juros_descontos",
  "locale": {
    "currency": "BRL"
  },
  "customer": {
    "firstName": "Name",
    "lastName": "Silva",
    "birthDate": "01/01/2001",
    "gender": "M",
    "type": "INDIVIDUAL",
    "email": "name.silva@gmail.com",
    "phones": [
      {
        "type": "MOBILE",
        "number": "123"
      }
    ],
    "document": "42715960434"
  },
  "payments": [
    {
      "totalValue": 1248.48,
      "totalValueWithRate": 1507.5,
      "rate": 1.99,
      "rateValue": 259.02,
      "paymentMethod": "CREDIT_CARD",
      "billingAddress": {
        "address": "Rua XPTO",
        "number": "1234",
        "complement": "apto 1234",
        "neighborhood": "XPTO",
        "postalCode": "13330130",
        "city": "Indaiatuba",
        "state": "SP",
        "country": "BRA",
        "type": "RESIDENTIAL"
      },
      "creditCard": {
        "installmentAmount": 18,
        "encryptedNumber": "6062920UAXW6689",
        "valuePerInstallment": 83.75,
        "maxCET": 33.18,
        "holder": {
          "name": "Juros"
        },
        "brand": "HIPERCARD",
        "expirationMonth": "06",
        "expirationYear": "2016"
      }
    },
    {
      "recipient": {
        "recipientPhone": "12981112233",
        "recipientDocument": "12981112233",
        "recipientEmail": "juros12@dezenbro.com.br",
        "type": "NORMAL",
        "address": {
          "address": "Rua XPTO",
          "number": "1234",
          "complement": "apto 1234",
          "neighborhood": "XPTO",
          "reference": "none",
          "postalCode": "13330130",
          "city": "Indaiatuba",
          "state": "SP",
          "country": "BRA",
          "type": "RESIDENTIAL"
        }
      },
      "itemsGroup": {
        "items": [
          {
            "offer": {
              "sellerId": "1",
              "skuId": 1178993,
              "price": {
                "BRL": 669
              }
            },
            "quantity": 2,
            "discountPrice": {
              "BRL": 602.1
            }
          },
          {
            "discounts": [
              {
                "rule": {
                  "id": 24274,
                  "type": "PERCENT",
                  "target": "ITEM",
                  "percentage": 10
                },
                "value": 66.9
              }
            ]
          },
          {
            "sla": {
              "id": "1",
              "type": "Normal",
              "duration": {
                "unit": "BUSINESS_DAY",
                "value": "35"
              },
              "price": {
                "BRL": 46.28
              }
            }
          }
        ]
      },
      "discounts": [
        {
          "rule": {
            "id": 24274,
            "type": "WOMINAL",
            "target": "ITEMS_GROUP",
            "nominal": 2
          },
          "value": 2
        }
      ]
    }
  ]
}
```

Campo	Descrição	Obrigatoriedade	Origem
genericOrderId	Identificador do pedido para quem está invocando a API (deve ser único para cada pedido gerado)	Sim	--
locale.currency	Moeda local (Brasil = BRL)	Sim	--
customer.firstName	Primeiro Nome do Cliente	Sim	--
customer.lastName	Último Nome do Cliente (sobrenome)	Sim	--
customer.birthDate	Data de Aniversário do Cliente	Não	--
customer.gender	Sexo do Cliente	Não	--
customer.type	Tipo do Cliente (exemplo: INDIVIDUAL)	Sim	--
customer.email	Email do Cliente	Sim	--
customer.phones[] .type	Tipo de Telefone (exemplo: MOBILE)	Sim	--
customer.phones[] .number	Número do Telefone do Cliente	Sim	--
customer.document	Número do Documento do Cliente	Sim	--
payments[] .totalValue	Valor Total do Pedido (se juros)	Sim	"discountPrice" * "quantity" (items) + "price" (do tipo de Entrega escolhida) + "Normal" ou "Agendada" - "discount" * ("desconto do carrinho, se houver")
payments[] .totalValueWithRate	Valor Total do Pedido Com Juros (se pagamento for parcelado o juros)	Somente se Pagamento for parcelado o juros	"totalValueWithRate" retornado na resposta do WM API /installments
payments[] .rate	Percentual do Juros Aplicado	Somente se Pagamento for parcelado o juros	"rate" retornado na resposta do WM API /installments
payments[] .rateValue	Valor do Juros	Somente se Pagamento for parcelado o juros	"rateValue" retornado na resposta do WM API /installments
payments[] .paymentMethod	Meio de Pagamento (exemplos: "CREDIT_CARD", "POST_PAID")	Sim	--
payments[] .billingAddress	Endereço do Faturamento	Sim para B2B2C, Opcional para modelo FI	Para Fidelidade e incentivo os dados de billingAddress (endereço de cobrança) não são utilizados e são opcionais
payments[] .billingAddress.address	Endereço do Faturamento (logradouro)	Sim	--
payments[] .billingAddress.number	Número (endereço)	Opcional	--
payments[] .billingAddress.complement	Complemento (endereço)	Opcional	--
payments[] .billingAddress.neighborhood	Bairro (endereço)	Sim	--
payments[] .billingAddress.postalCode	CEP (endereço)	Sim	--
payments[] .billingAddress.city	Cidade (endereço)	Sim	--
payments[] .billingAddress.state	Estado (endereço)	Sim	--
payments[] .billingAddress.country	País (endereço)	Sim	--
payments[] .billingAddress.type	Tipo de Endereço (Exemplo: "RESIDENTIAL")	Sim	--
payments[] .creditCard.installmentAmount	Quantidade de Parcelas	Somente se Meio de Pagamento for em Cartão de Crédito	"installmentAmount" retornado na resposta do WM API /installments
payments[] .creditCard.encryptedNumber	Número do Cartão de Crédito Encriptografado	Somente se Meio de Pagamento for em Cartão de Crédito	"encryptedNumber" retornado na resposta do WM API /tokenize
payments[] .creditCard.valuePerInstallment	Valor por Parcela	Somente se Meio de Pagamento for em Cartão de Crédito	"valuePerInstallment" retornado na resposta do WM API /installments
payments[] .creditCard.maxCET	Máximo do Custo Efetivo Total	Somente se Meio de Pagamento for em Cartão de Crédito	"maxCET" retornado na resposta do WM API /installments
payments[] .creditCard.holder.name	Nome do Titular do Cartão de Crédito	Somente se Meio de Pagamento for em Cartão de Crédito	--
payments[] .creditCard.brand	Bandeira do Cartão de Crédito	Somente se Meio de Pagamento for em Cartão de Crédito	retornado na resposta do WM API /paymentMethods
payments[] .creditCard.expirationMonth	Mês de Vencimento do Cartão de Crédito	Somente se Meio de Pagamento for em Cartão de Crédito	--
payments[] .creditCard.expirationYear	Ano de Vencimento do Cartão de Crédito	Somente se Meio de Pagamento for em Cartão de Crédito	--
shipping.recipient	Destinatário da Entrega	Sim	--
shipping.recipientPhone	Telefone do Destinatário da Entrega	Opcional	--
shipping.recipientDocument	Número do Documento do Destinatário da Entrega	Opcional	--
shipping.recipientEmail	Email do Destinatário da Entrega	Opcional	--
shipping.type	Tipo do Documento do Destinatário de Entrega (Exemplo: "NORMAL")	Opcional	--
shipping.address.address	Endereço de Entrega	Sim	--
shipping.address.number	Número (Endereço de Entrega)	Opcional	--
shipping.address.complement	Complemento (Endereço de Entrega)	Opcional	--
shipping.address.neighborhood	Bairro (Endereço de Entrega)	Sim	--
shipping.address.reference	Referência (Endereço de Entrega)	Opcional	--
shipping.address.postalCode	CEP (Endereço de Entrega)	Sim	--
shipping.address.city	Cidade (Endereço de Entrega)	Sim	--
shipping.address.state	Estado (Endereço de Entrega)	Sim	--
shipping.address.country	País (Endereço de Entrega)	Sim	--
shipping.address.type	Tipo de Endereço (Exemplo: "RESIDENTIAL")	Sim	--
itemsGroup.items[] .offer.sellerId	ID do Seller	Sim	"sellerId" retornado na resposta do WM API /simulate
itemsGroup.items[] .offer.skuId	ID do SKU	Sim	"skuId" retornado na resposta do WM API /simulate
itemsGroup.items[] .offer.price.BRL	Preço de Oferta	Sim	"price" retornado na resposta do WM API /simulate
itemsGroup.items[] .quantity	Quantidade do Item	Sim	"quantity" retornado na resposta do WM API /simulate
itemsGroup.items[] .discountPrice.BRL	Preço do Item Com Desconto	Sim	"discountPrice" retornado na resposta do WM API /simulate
itemsGroup.items[] .discounts[] .rule.id	ID do Desconto (se houver)	Somente se houver Desconto na Simulação (WM API /simulate)	"id" (itemsGroup.items[] discounts[] rule) retornado na resposta do WM API /simulate
itemsGroup.items[] .discounts[] .rule.type	Tipo do Desconto (Exemplos: "PERCENT", "FIXED")	Somente se houver Desconto na Simulação (WM API /simulate)	"type" (itemsGroup.items[] discounts[] rule) retornado na resposta do WM API /simulate
itemsGroup.items[] .discounts[] .rule.target	Alvo de Desconto (Exemplos: "ITEM", "ITEMS_GROUP", "FREIGHT")	Somente se houver Desconto na Simulação (WM API /simulate)	"target" (itemsGroup.items[] discounts[] rule) retornado na resposta do WM API /simulate
itemsGroup.items[] .discounts[] .rule.percentage	Percentual do Desconto Aplicado no Item	Somente se houver Desconto na Simulação (WM API /simulate)	"percentage" (itemsGroup.items[] discounts[] rule) retornado na resposta do WM API /simulate
itemsGroup.items[] .discounts[] .value	Valor do Percentual de Desconto	Somente se houver Desconto na Simulação (WM API /simulate)	"value" (itemsGroup.items[] discounts[] rule) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.id	ID do SLA (Entrega)	Sim	"id" (itemsGroup.items[] delivery.sla[]) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.type	Tipo da Entrega (Exemplo: "Normal", "Agendada")	Sim	"type" (itemsGroup.items[] delivery.sla[]) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.duration.unit	Unidade de duração de SLA (Exemplos: "BUSINESS_DAY", "DAY", "WEEK", "HOUR", "DATE")	Sim	"unit" (itemsGroup.items[] delivery.sla[] duration) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.duration.value	Valor da Unidade de duração da Entrega. Exemplo: 22 (dias)	Sim	"value" (itemsGroup.items[] delivery.sla[] duration) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.price.BRL	Valor da Entrega (frete)	Sim	"BRL" (itemsGroup.items[] delivery.sla[] price) retornado na resposta do WM API /simulate * esse valor já é retornado de acordo com a quantidade de itens simulados!
discounts[] .rule.id	ID do Desconto de Carrinho (NOMINAL, se houver)	Somente se houver Desconto de Carrinho (NOMINAL) na Simulação (WM API /simulate)	"id" (discount[] rule) retornado na resposta do WM API /simulate
discounts[] .rule.type	Tipo do Desconto de Carrinho (Exemplo: "NOMINAL")	Somente se houver Desconto de Carrinho (NOMINAL) na Simulação (WM API /simulate)	"type" (discount[] rule) retornado na resposta do WM API /simulate
discounts[] .rule.target	Alvo de Desconto de Carrinho (Exemplo: "ITEMS_GROUP")	Somente se houver Desconto de Carrinho (NOMINAL) na Simulação (WM API /simulate)	"target" (discount[] rule) retornado na resposta do WM API /simulate
discounts[] .rule.nominal	Desconto do Valor Nominal (carrinho)	Somente se houver Desconto de Carrinho (NOMINAL) na Simulação (WM API /simulate)	"nominal" (discount[] rule) retornado na resposta do WM API /simulate
discounts[] .value	Valor do Desconto Nominal (carrinho)	Somente se houver Desconto de Carrinho (NOMINAL) na Simulação (WM API /simulate)	"value" (discount[] rule) retornado na resposta do WM API /simulate

Exemplo de Entrega Agendada

POST/orders - Response Exemplo

Expand

source

```
{
  "itemsGroup": {
    "items": [
      {
        "offer": {
          "sellerId": "1",
          "sku": 34542,
          "price": {
            "BRL": 655.50
          }
        },
        "quantity": 1,
        "discountPrice": {
          "BRL": 655.50
        },
        "sla": {
          "id": "5",
          "type": "Agendada",
          "duration": {
            "unit": "BUSINESS_DAY",
            "value": "18"
          },
          "price": {
            "BRL": 15.00
          }
        },
        "schedule": {
          "begin": "2015-12-29T07:00:00-02:00",
          "end": "2015-12-29T12:00:00-02:00",
          "price": {
            "BRL": 15.00
          }
        }
      }
    ]
  }
}
```

Campo	Descrição	Obrigatoriedade	Origem
itemsGroup.items[] .sla.id	ID do SLA (da Entrega Agenda, Exemplo: "5")	Sim	"id" (itemsGroup.items[] delivery.sla) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.type	Tipo da Entrega, no caso: "Agendada"	Sim	"type" (itemsGroup.items[] delivery.sla) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.duration.unit	Unidade de duração de SLA (Exemplos: "BUSINESS_DAY", "DAY", "WEEK", "HOUR", "DATE")	Sim	"unit" (itemsGroup.items[] delivery.sla) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.duration.value	Valor da Unidade de duração da Entrega Agendada, Exemplo: 18 (dias)	Sim	"value" (itemsGroup.items[] delivery.sla) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.price.BRL	Valor da Entrega Agendada (frete)	Sim	"BRL" (itemsGroup.items[] delivery.sla) retornado na resposta do WM API /simulate * esse valor já é retornado de acordo com a quantidade de itens simulados!
itemsGroup.items[] .sla.schedule.begin	Data inicial da Entrega Agendada (Exemplo: "2015-12-29T07:00:00-02:00")	Sim	"begin" (itemsGroup.items[] delivery.sla) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.schedule.end	Data final da Entrega Agendada (Exemplo: "2015-12-29T12:00:00-02:00")	Sim	"end" (itemsGroup.items[] delivery.sla) retornado na resposta do WM API /simulate
itemsGroup.items[] .sla.schedule.price.BRL	Valor da Entrega Agendada (frete) * Já considerando a quantidade informada	Sim	"BRL" (itemsGroup.items[] delivery.sla) retornado na resposta do WM API /simulate

/orders/tracking/{id}

Considerações:

- **Payment:** O campo **totalValueWithRate** somente será retornado caso o parcelamento possua juros.
- Para modelo **Fidelidade Incentivo** os campos **billingAddress** e **recipientEmail** são retornados NULL.

Response

GET/orders/tracking/{id} - Response Exemplo

Expand

source

```
{
  "summary": {
    "statusCode": 200
  },
  "tracking": {
    "orderId": "51234192",
    "genericOrderId": "388411446746180218442433",
    "customer": {
      "firstName": "Elaine",
      "lastName": "Duarte",
      "email": "elaine@ciandt.com",
      "phones": [
        {
          "type": "MOBILE",
          "number": "19988343843"
        },
        {
          "type": "HOME",
          "number": "1988888887"
        }
      ],
      "gender": null,
      "document": "38841442433"
    },
    "orderStatusList": [
      {
        "status": "ORC",
        "date": "2016-04-25T18:34:00.793-0300",
        "description": "Order created"
      },
      {
        "status": "ORCL",
        "date": "2016-04-25T18:37:01.001-0300",
        "description": "Order created in legacy"
      },
      {
        "status": "ORA",
        "date": "2016-04-25T18:38:12.001-0300",
        "description": "Order risk approved"
      },
      {
        "status": "OWPC",
        "date": "2016-04-25T18:38:12.002-0300",
        "description": "Order waiting payment callback"
      },
      {
        "status": "OPS",
        "date": "2016-04-25T19:46:25.001-0300",
        "description": "Order paid"
      }
    ],
    "orderTotal": {
      "BRL": 2568.97
    },
    "itemsTotal": {
      "BRL": 2488.01
    },
    "freightTotal": {
      "BRL": 80.96
    },
    "itemsGroup": {
      "items": [
        {
          "offer": {
            "skuId": "34665",
            "sellerId": "1",
            "unit": "1",
            "quantity": 4,
            "discountPrice": {
              "BRL": 0
            },
            "sla": {
              "id": "1",
              "type": "Normal",
              "duration": {
                "unit": "BUSINESS_DAY",
                "value": 26
              },
              "schedule": null,
              "price": {
                "BRL": 26
              }
            },
            "shipping": {
              "recipient": "Elaine Duarte",
              "recipientPhone": "88888887",
              "recipientEmail": null,
              "address": {
                "address": "Rua Francisca Fátima de Oliveira",
                "number": "136",
                "complement": null,
                "neighborhood": "Jardim Nossa Senhora Auxiliadora",
                "postalCode": "13183480",
                "city": "Hortolândia",
                "state": "SP",
                "country": "BRA",
                "type": "RESIDENTIAL"
              }
            }
          },
          "offer": {
            "skuId": "895123",
            "sellerId": "mobly"
          },
          "quantity": 1,
          "discountPrice": {
            "BRL": 139.99
          },
          "sla": {
            "id": "Normal",
            "type": "Normal",
            "duration": {
              "unit": "BUSINESS_DAY",
              "value": 26
            },
            "schedule": null,
            "price": {
              "BRL": 18.96
            }
          },
          "shipping": {
            "recipient": "Elaine Duarte",
            "recipientPhone": "88888887",
            "recipientEmail": null,
            "address": {
              "address": "Rua Francisca Fátima de Oliveira",
              "number": "136",
              "complement": null,
              "neighborhood": "Jardim Nossa Senhora Auxiliadora",
              "postalCode": "13183480",
              "city": "Hortolândia",
              "state": "SP",
              "country": "BRA",
              "type": "RESIDENTIAL"
            }
          }
        },
        {
          "offer": {
            "skuId": "29134",
            "sellerId": "1",
            "unit": "1",
            "quantity": 1,
            "discountPrice": {
              "BRL": 2218.12
            },
            "sla": {
              "id": "1",
              "type": "Normal",
              "duration": {
                "unit": "BUSINESS_DAY",
                "value": 26
              },
              "schedule": null,
              "price": {
                "BRL": 49
              }
            },
            "shipping": {
              "recipient": "Elaine Duarte",
              "recipientPhone": "88888887",
              "recipientEmail": null,
              "address": {
                "address": "Rua Francisca Fátima de Oliveira",
                "number": "136",
                "complement": null,
                "neighborhood": "Jardim Nossa Senhora Auxiliadora",
                "postalCode": "13183480",
                "city": "Hortolândia",
                "state": "SP",
                "country": "BRA",
                "type": "RESIDENTIAL"
              }
            }
          },
          "offer": {
            "skuId": "120611",
            "sellerId": "1",
            "unit": "1",
            "quantity": 1,
            "discountPrice": {
              "BRL": 129.9
            },
            "sla": {
              "id": "1",
              "type": "Normal",
              "duration": {
                "unit": "BUSINESS_DAY",
                "value": 26
              },
              "schedule": null,
              "price": {
                "BRL": 13
              }
            },
            "shipping": {
              "recipient": "Elaine Duarte",
              "recipientPhone": "88888887",
              "recipientEmail": null,
              "address": {
                "address": "Rua Francisca Fátima de Oliveira",
                "number": "136",
                "complement": null,
                "neighborhood": "Jardim Nossa Senhora Auxiliadora",
                "postalCode": "13183480",
                "city": "Hortolândia",
                "state": "SP",
                "country": "BRA",
                "type": "RESIDENTIAL"
              }
            }
          }
        }
      ]
    },
    "payments": [
      {
        "totalValue": {
          "BRL": 2568.97
        },
        "paymentMethod": "CREDIT_CARD",
        "billingAddress": {
          "address": "Rua Francisca Fátima de Oliveira",
          "number": "136",
          "complement": null,
          "neighborhood": "Jardim Nossa Senhora Auxiliadora",
          "postalCode": "13183480",
          "city": "Hortolândia",
          "state": "SP",
          "country": "BRA",
          "type": "RESIDENTIAL"
        },
        "creditCard": {
          "installmentAmount": 7,
          "holder": {
            "name": "Elaine Duarte"
          },
          "brand": "HIPERCARD",
          "valuePerInstallment": {
            "BRL": 394.58
          },
          "maxCET": 29.13
        },
        "totalValueWithRate": {
          "BRL": 2762.06
        },
        "rateValue": {
          "BRL": 193.09
        },
        "rate": 1.7
      }
    ],
    "deliveries": [
      {
        "carrier": {
          "id": "19632103000100",
          "name": "RODOWEB TRANSPORTADORA LTDA",
          "type": "RODOWEB"
        },
        "outputDate": null,
        "estimatedDate": "2015-12-14T00:00:00.000-0200",
        "pastEstimatedDate": "2015-11-17T00:00:00.000-0200",
        "deliveryDate": null,
        "logisticType": "DELIVERY",
        "deliveryItems": [
          {
            "offer": {
              "sellerId": "1",
              "skuId": "34665"
            },
            "quantity": 4
          }
        ],
        "deliveryStatusList": [],
        "invoices": []
      },
      {
        "carrier": {
          "id": "34028316003129",
          "name": "TESTE TRANSP 1",
          "type": "TESTELOCK1"
        },
        "outputDate": null,
        "estimatedDate": "2015-11-11T00:00:00.000-0200",
        "pastEstimatedDate": "2015-11-17T00:00:00.000-0200",
        "deliveryDate": null,
        "logisticType": "DELIVERY",
        "deliveryItems": [
          {
            "offer": {
              "sellerId": "1",
              "skuId": "120611"
            },
            "quantity": 1
          }
        ],
        "deliveryStatusList": [
          {
            "status": "DCA",
            "date": "2016-05-13T12:07:50.004-0300",
            "description": "Delivery Canceled"
          }
        ],
        "invoices": []
      }
    ]
  }
}
```

GET/orders/tracking/{id} - Response Erro

Expand

source

```
{
  "summary": {
    "statusCode": 404,
    "errors": [
      {
        "type": "NOT_FOUND",
        "message": "Not Found"
      }
    ]
  }
}
```

GET/orders/tracking/{id} - Response Schema

Expand

source

```
TrackingResponse {
  summary (Summary),
  tracking (TrackingBody, optional)
}

Summary {
  statusCode (integer),
  errors (Array[SummaryErrors], optional): Errors list
}

TrackingBody {
  orderId (integer, optional),
  genericOrderId (string, optional): Generic Order Id,
  customer (TrackingCustomer, optional),
  orderStatusList (Array[TrackingStatus], optional): Order Status List,
  orderTotal (object, optional),
  itemsTotal (object, optional),
  freightTotal (object, optional),
  itemsGroup (TrackingItemsGroup, optional),
  payments (Array[TrackingPayment], optional),
  deliveries (Array[TrackingDelivery], optional)
}

SummaryErrors {
  message (string, optional),
  type (string, optional)
}

TrackingCustomer {
  firstName (string, optional): First Name,
  lastName (string, optional): Last Name,
  gender (string, optional): Gender,
  type (string, optional): Customer type,
  email (string, optional): Email,
  phones (Array[TrackingPhone], optional): Phones,
  document (string, optional): Document
}

TrackingStatus {
  status (string, optional): Tracking status code,
  description (string, optional): Tracking status description,
  date (string, optional): Order status date
}

TrackingItemsGroup {
  items (Array[TrackingItem], optional)
}

TrackingPayment {
  totalValue (object, optional): Total value,
  valuePerInstallment (object, optional): Value per installment,
  rate (number, optional): Percentual interest,
  rateValue (object, optional): Interest value,
  paymentMethod (string, optional): Payment Method,
  billingAddress (TrackingAddress, optional): Billing Address,
  creditCard (TrackingCreditCard, optional): Credit Card
}

TrackingDelivery {
  carrier (TrackingCarrier, optional),
  outputDate (string, optional): Output date,
  estimatedDate (string, optional): Estimated date,
  pastEstimatedDate (string, optional): Past Estimated date,
  deliveryDate (string, optional): Delivery date,
  logisticType (string, optional): Logistic type,
  deliveryItems (Array[TrackingDeliveryItem], optional),
  deliveryStatusList (Array[TrackingStatus], optional),
  invoices (Array[TrackingInvoices], optional)
}

TrackingPhone {
  type (string, optional): Phone type,
  number (string, optional): Phone number
}

TrackingItem {
  offer (TrackingOffer, optional),
  quantity (integer, optional): Item quantity,
  discountPrice (object, optional),
  sla (TrackingSla, optional),
  shipping (TrackingShipping, optional)
}

TrackingAddress {
  address (string, optional): Destinatary address,
  number (string, optional): Destinatary address number,
  neighborhood (string, optional): Destinatary address complement,
  neighborhood (string, optional): Destinatary address neighborhood,
  postalCode (string, optional): Destinatary address postal code,
  city (string, optional): Destinatary address city,
  state (string, optional): Destinatary address state,
  country (string, optional): Destinatary address country,
  type (string, optional): Destinatary address type
}

TrackingCreditCard {
  installmentAmount (integer, optional): Installment Amount,
  valuePerInstallment (object, optional): Value per installment,
  maxCET (number, optional): Max CET(Custo Efetivo Total),
  holder (TrackingCardHolder, optional): Holder,
  brand (string, optional): Credit card brand
}

TrackingCarrier {
  id (string, optional): Carrier id,
  name (string, optional): Carrier name,
  type (string, optional): Carrier type
}

TrackingDeliveryItem {
  offer (TrackingOffer, optional),
  quantity (integer, optional): Item quantity
}

TrackingInvoices {
  series (number, optional),
  nfeAccessKey (string, optional),
  issuerDocNumber (number, optional),
  number (number, optional)
  * % Os atributos BillingAddress e recipientEmail serão retornados como NULL para usuários no modelo FI
}

TrackingOffer {
  sellerId (string, optional): Seller Id,
  skuId (integer, optional): Sku Id
}

TrackingSla {
  id (string, optional),
  type (string, optional),
  duration (TrackingDuration, optional),
  schedule (TrackingSchedule, optional): only if is scheduled delivery type,
  price (object, optional)
}

TrackingShipping {
  recipient (string, optional): Destinatary name,
  recipientPhone (string, optional): Destinatary phone,
  recipientEmail (string, optional): Destinatary email,
  address (TrackingAddress, optional): Shipping Address
}

TrackingCardHolder {
  name (string, optional): Holder name
}

TrackingDuration {
  value (string, optional),
  unit (string, optional)
}

TrackingSchedule {
  begin (string, optional): Initial date,
  end (string, optional): End date
}
```

Status de tracking

Lista de status de pedido:

Status	Descrição
ORC	Pedido criado.
ORCL	Pedido criado no legado Sige.
ONP	Pedido não pago.
OWDP	Aguardando entregas pagamento.
OWSR	Pedido aguardando recuperação de vendas.
OWPR	Pedido aguardando a troca de meio de pagamento do pedido.
ORD	Pedido negado na análise de risco.
ORA	Pedido aprovado na análise de risco.
OWPC	Pedido aguardando callback do pagamento.
OPS	Pedido pago com sucesso.

Lista de status de entrega:

Status	Descrição
EII	Fatura eletrônica emitida.
II	Fatura emitida.
SWB	Venda sem estoque.
GRCB	Mercadoria recebida na base do transportador.
DLC	Entrega criada.
DR	Entrega reagendada.
DWR	Entrega aguardando reagendamento.
WWA	Aguardando pro retirada na Agência.
PD	Dano parcial.
TD	Avaria Total.
PL	Perda parcial.
TL	Perda total.
TBBC	Transferência entre Filiais da Transportadora.
IDR	Em Rota de Entrega.
CCA	Cliente destino mudou de endereço.
CANF	Endereço do cliente destino não localizado.
LDL	Entregue.
IRRWMB	Em rota de devolução ao WMB.
WDA	Endereço de entrega errado.
CRD	Entrega recusada pelo Cliente.
EC	Estabelecimento fechado.
CICM	Faltando complemento do endereço do cliente.
ENAR	Funcionário não autorizado a Receber Mercadorias.
GDWO	Mercadoria em Desacordo com o Pedido Compra.
RRA1	Responsável pelo recebimento ausente 1.
RRA2	Responsável pelo recebimento ausente 2.
RRA3	Responsável pelo recebimento ausente 3.
CT	Carga roubada.
RWD	Recusa por Deficiência Embalagem Mercadoria.
FSE	Selagem fiscal.
DS	Triagem Diferenciada.
NCS	Entrega não confirmada
DCA	Entrega Cancelada

Padrões de Resposta / Rastreabilidade

As informações a seguir são aplicáveis a todas as APIs.

Os seguintes códigos de resposta são retornados:

HTTP Status	Descrição
200	Sucesso na execução da operação
400	Erro de validação de um ou mais parâmetros de entrada
401	Acesso não autorizado. A requisição requer autenticação
403	Requisição negada por falta de permissão para executar a operação
404	O recurso requisitado não existe
405	Método não Permitido
410	O recurso requisitado não está mais ativo
500	Erro interno da aplicação na execução da operação

Padrão para Datas

Considere o padrão ISO-8601, para todos atributos de data.

Ex: 2015-10-28T12:00:00-0200

Padrão de Retorno: Sumário

Todas requisições efetuadas na API, tem em sua resposta um atributo "summary", que fornece um sumário com detalhes da resposta:

erroId: Código de erro gerado pelo sistema, que indica o tipo de erro.

errors: Lista de erros ocorridos na requisição. Contém os atributos **"message"**, com uma mensagem descritiva do erro, e caso aplicável, fornece um código correspondente ao erro no atributo **"type"**.

```
"summary": {
  "statusCode": 404,
  "errors": [
    {
      "message": "Requested resource not found"
      "type": "NOT_FOUND"
    }
  ]
},
```

Rastreabilidade

As aplicações que consumirém os serviços serão responsáveis em manter a rastreabilidade da informação para facilitar a monitoração e análise de LOGs.

Para isso os serviços expostos fornecem o parâmetro "transactionId", que gera um identificador único da transação. Uma vez gerado esse identificador, é desejável que para as requisições seguintes que fizerem parte da mesma operação de compra (mesma sessão de usuário), esse parâmetro seja fornecido na requisição, para que um novo identificador da transação não seja gerado.

Exemplo:

```
"transactionId": 912FF3090132A89D12
```

Changelog

Abaixo é descrito o histórico de versões da API:

[X.X.X] - 01/09/2016 (Versão Atual - a partir de 01/09/2016)

Modificado

- Alterado obrigitoriedade de campos para **place/order** de pedidos na modalidade Fidelidade Incentivo (FI):
 - O atributo **billingAddress** passa a ser **opcional** para pedidos de usuários no modelo **FI**, não sendo utilizado no processo.
- Alterado retorno de atributos no **tracking** de pedidos na modalidade Fidelidade Incentivo (FI):
 - Não pode conter somente números.
 - Código internacional é opcional.
 - Exemplos válidos: +5519999999999, 5519999999999, 19999999999 e 19999999999.
- Sugestão de preenchimento do atributo **"shipping"** adicionado na documentação da API do Orders.

[4.3.68] - 05/07/2016 (Versão até 01/09/2016)

Modificado

- Alterado o mapeamento de validações dos atributos do **/simulate** e **/orders** via swagger.
- Telefones:
 - customer.phones.number e shipping.recipientPhone.
 - Tamanho mínimo: 10 e tamanho máximo: 14.
 - Formato aceito: Código internacional (+55 ou +56) + DDD (2 dígitos) + Telefone (8 ou 9 dígitos).
 - Somente valores numéricos.
 - Código internacional é opcional.
 - Exemplos válidos: +5519999999999, 5519999999999, 19999999999 e 19999999999.
- CEP:
 - payments.billingAddress.postalCode e shipping.address.postalCode.
 - Tamanho máximo: 10.
 - Somente valores numéricos.
 - Exemplo: 13490000.
- Número do endereço:
 - shipping.address.number e payments.billingAddress.number.
 - Tamanho máximo: 6.
 - Somente valores numéricos.
 - Exemplo: 123456.
- Nome Destinatário:
 - shipping.recipient.
 - Tamanho máximo: 100.
 - Não pode conter somente números.
 - Exemplo: José da Silva.
- Nome Destinatário:
 - customer.firstName e customer.lastName.
 - Tamanho máximo: 50.
 - Não pode conter somente números.
 - Exemplo: João da Silva.
- Email:
 - shipping.recipientEmail e customer.email.
 - Tamanho máximo: 100.
 - Pattern: /^[A-Za-z0-9+!{}~|_](.[A-Za-z0-9+!{}~|_]{1,64})?@[A-Za-z0-9+!{}~|_](.[A-Za-z0-9+!{}~|_]{1,64})?\$/
 - Exemplo: j.palmer@walmart.com.
- Nome Credit Card Holder:
 - payments.creditCardHolder.name.
 - Tamanho máximo: 50.
 - Não pode conter somente números.
- Documento Credit Card Holder:
 - payments.creditCardHolder.document.
 - Tamanho mínimo: 11.
 - Tamanho máximo: 14.
 - Deve conter somente números.
 - Aceita somente 11 ou 14 números.
 - Pode ser nulo.
- Documento de destinatário:
 - shipping.recipientDocument.
 - Tamanho mínimo: 11.
 - Tamanho máximo: 14.
 - Deve conter somente números.
 - Aceita somente 11 ou 14 números.
 - Pode ser nulo.

[4.3.65] - 23/06/2016

Adicionado

- Funcionalidade para efetuar pedidos sem a necessidade de um cadastro de cliente para pedidos FI (Fidelidade e Incentivo).

[4.3.56] - 06/06/2016

Adicionado

- Funcionalidade para efetuar pedidos sem a necessidade de um cadastro de cliente para pedidos B2B2BC.
- Funcionalidade para permitir Valores e Descontos de Frete específicos para a Plataforma B2B

Alterado

- Nos Catálogos Full e Partial, apenas os itens aptos para os Sellers selecionados por Usuário são retornados. Atualmente todos os parceiros recebem apenas itens Walmart.

- Nos endpoints /products e /skus apenas itens para o Sellers selecionados por Usuário são retornados: Não temos alterações de contrato, apenas do conteúdo retornado.
- No endpoint /order apenas itens para o Sellers selecionados por Usuário são permitidos. Não temos alterações de contrato, apenas das validações realizadas.
- No endpoint /order foi incluído o OrderID existente nos detalhes do erro quando é realizada uma tentativa de place com GenericID existente. Vide documentação do Orders API para mais detalhes.

Segurança

- Tratamento com Validade do Cartão dentro PCI e análise anti-fraude.

Correções

- Tratamento nas mensagens de retorno do Tracking quando é solicitado um OrderId Inválido.

[4.3.56] - 19/05/2016

- Alterado o mapeamento de status do Tracking, a partir de agora novos status de Entrega Cancelada serão exibidos. Vide documentação do Orders API para mais detalhes.

Modificado

- Alterado o mapeamento de status do tracking, a partir de agora novos status de entrega serão exibidos.

[4.3.56] - 02/05/2016

Modificado

- Alterado o mapeamento de status do tracking, a partir de agora novos status de entrega serão exibidos.

[4.3.56] - 11/04/2016

Modificado

- Nos endpoints /products e /skus, a oferta contida no atributo "buyBox" também aparecerá na lista "offers" do "standardSKU"
- Incluído EAN nos detalhes dos endpoints /products e /skus,
- Incluído EAN nos catálogos de produto.
- Alterado listagem de status apresentadas no tracking de pedidos. Vide documentação do Orders API para mais detalhes.

[4.3.52] - 17/03/2016

Adicionado

- API passa a suportar fluxo de negócio Fidelidade Incentivo.
- Parceiros da modalidade Fidelidade Incentivo terão catálogos contendo apenas produtos Walmart e somente meio de pagamento pós pago.
- Suporte a meio de pagamento pós pago (POST_PAID). Nesses casos o objeto creditCard não deverá ser passado.
- Para clientes da modalidade Fidelidade Incentivo só será aceito o meio de pagamento pós pago "POST_PAID".
- Suporte para buscar todas as formas de parcelamento sem juros disponíveis.
- Adicionado bloqueio dos endpoints /installments e /tokenizer para os para usuários da modalidade Fidelidade Incentivo (FI).

[4.3.41] - 25/02/2016

Corrigido

- API passa a suportar que atributos opcionais da requisição sejam enviados com valor "null". Anteriormente caso se desejasse não passar um atributo opcional, devia-se omiti-lo da requisição.
- Preço da entrega agendada com valor corrigido, ex. 790 para 7.90.

[4.3.38] - 18/02/2016

Segurança

- Modificações no serviço de autenticação da API, para adequações relativas ao padrão OAuth 2.0 [com modificações do contrato]

[4.3.65] - 23/06/2016 (Versão atual)