Parcelow – Payments API

The objective of this document is to guide the developers on the steps to integrate the **Parcelow Payments API**, describe its functionalities and methods that are going to be used, list the information to be sent and received, and provide examples.

The integration mechanism with the Payment API is simple. Therefore, in order to successfully implant the solutions, a programmer with intermediate skills in web programming language, HTTP/HTTPS requisitions, and JSON archive handling, will be able to perform them.

This manual will refer to all the available operations in the API REST of the API Payments. These operations must be executed using a specific key (Client ID and Client Secret) of the corresponding surrounding endpoints:

**Sandbox**: <https://sandbox.parcelow.com/>

**Production**: <https://app.parcelow.com/>

# Customer Access Registration

Para ter acesso à API, primeiramente deve-se solicitar por e-mail o cadastramento de seu client.

É necessário que o sistema que está integrando tenha uma *Account* já ativa no sistema Parcelow.

Ao solicitar o cadastramento do client é necessário enviar uma URL padrão para receber notificações POST (webhook) com a atualização dos pedidos feitos através da API.

Então você receberá o **client\_id** e o **client\_secret** que devem ser guardados em segurança e utilizados para autenticação na API.

In order to have access to the API, first, you must email a request to sign up a customer.

The system you are integrating must have an active account in the Parcelow system.

When you make the customer registration request, it is necessary to send a URL template to receive POST (webhook) notifications with an update of the orders made through the API.

After that, you will receive the **client\_id** and **client\_secret**, which should be kept securely, and they will be used for the API authentication.

# Request Access Token

Before you can request to create or query transactions you must obtain an API access token.

This is done by making a request to the API with your client\_id and client\_secret received when requesting your client registration.

|  |  |  |  |
| --- | --- | --- | --- |
| **Request:** | **POST** /oauth/token | | |
| **Properties** | **Type** | **Mandatory** | **Description** |
| client\_id | text | yes | Client ID received when registration to API is requested |
| client\_secret | text | yes | Client Secret received at API registration |
| grant\_type | text | yes | Fixed: “client\_credentials" |

|  |
| --- |
| **Example** |
| curl --request POST 'https://sandbox.parcelow.com/oauth/token' \  --form 'client\_id=1' \  --form 'client\_secret=TCJhyj1wgunOS7CvoF0Q1va2Z04KUhID3H9JqFiq' \  --form 'grant\_type=client\_credentials' |

|  |
| --- |
| **Return** |
| {  "token\_type": "Bearer",  "expires\_in": 31622400,  "access\_token": "eAiOiJKV1QiLCJhbGc[...]SUzI1NiIsI5MjU1MDM3N"  } |

# Create Transaction

Endpoint used to create your transactions in the API. The call must be authenticated with the Access TOKEN created in the previous step.

Observation 1: All the amounts must be in US$

Observation 2: Shipping is optional

Observation 3: The parameters can be sent in JSON in the body of the request or in form-data format (POST parameters)

Observation 4: The redirecting URLs must respond to status 200 to be considered valid.

Observation 5: The payment link generated at the request is good for 10 days. You can use the endpoint of the transaction inquiry to obtain a new updated link.

|  |  |  |  |
| --- | --- | --- | --- |
| **Request:** | **POST** /api/orders | | |
| **Headers:** | Content-Type | | application/json |
| Authorization | | Bearer: {token} |
| **Properties** | **Type** | **Mandatory** | **Description** |
| reference | text | no | field for free use of the integrating system |
| disable\_email\_notifications | boolean | no | Field to define whether e-mail notifications about the transaction will be sent. |
| client | key | no | personal information of paying customer |
| client.cpf | text | no | CPF (Brazilian SSN) of paying customer. Just numbers |
| client.name | text | no | name of paying customer |
| client.email | text | no | contact email of paying customer |
| client.birthdate | text | no | birth date of paying customer |
| client.cep | text | no | Brazilian zipcode of paying customer’s address |
| client.phone | text | no | contact phone number of paying customer |
| items | key | yes | Item description of the future order. It can have n items |
| items[0].reference | text | no | field for free use of the integrating system |
| items[0].description | text | yes | description of the item being purchased. |
| items[0].quantity | integer | yes | quantity of items to be purchased |
| items[0].amount | integer | yes | unit price of items to be purchased, in cents |
| shipping | key | no | Optional. Shipping Array information |
| shipping.amount | integer | no | shipping costs in cents. |
| coupon.code | text | no | Coupon code used at checkout |
| coupon.value | interger | no | Discount amount in cents |
| redirect | key | yes | redirecting URLs Array |
| redirect.success | text | yes | return URL in case of successful payment. |
| redirect.failed | text | yes | return URL in case of unsuccessful payment |

|  |
| --- |
| **Example** |
| curl --request POST 'https://sandbox.splipay.com/api/orders' \  --header 'Content-Type: application/json' \  --header 'Authorization: Bearer eyJ0eXAiOiJ.eyJnRpIjo.rLHS2' \  --data-raw '{  "reference": "my-order-ref-0001",  "client": {  "cpf": "43164853858",  "name": "John Doe",  "email": "johndoe@example.com",  "birthdate": "1982-01-14",  "cep": "18053190",  "phone": "15987452584"  },  "items": [  {  "reference": "my-item-ref-0001",  "description": "Apple iMac 21.5-inch",  "quantity": "1",  "amount": 149900  },  {  "reference": "my-item-ref-0002",  "description": "Apple iPhone 11 Pro",  "quantity": 1,  "amount": 99900  }  ],  "coupon": {  "code": "10DESCONTO",  "value": 1000,  "issuer": "merchant\_api"  },  "shipping": {  "amount": 1000  },  "redirect": {  "success": "https://example.com/sp\_callback?success",  "failed": "https://example.com/sp\_callback?failed"  }  }' |

|  |
| --- |
| **Return** |
| {  "success": true,  "data": {  "order\_id": 57,  "url\_checkout": "https://sandbox.parcelow.com/payment/4openRe7Az/kzPdyP7bQr?expires=1583164527&signature=e1caae419137903f930f5a5b3d4c72608a61c0fdd36c70803d4a92c15556a4c5"  }  } |

|  |
| --- |
| **Return Error: HTTP 400 status** |
| {  "message": "The given data was invalid.",  "errors": {  "client.birthdate": [  "The client.birthdate field is required."  ]  }  } |

# Personal Information Chalenge (Confirmation)

After creating the transaction, you can redirect the user to the included URL in the field “url\_checkout" in the previous answer.

From this point on, it is also possible to continue in Transparent Checkout mode. In this scenario, before processing the credit card payment, it is necessary to confirm the customers’ personal information.

Send a request with some security questions and answers. These questions must be shown to the customers, and the answers must be resent to the API to process and validate the users' personal information.

**How to get personal questions?**

|  |  |  |
| --- | --- | --- |
| **Request:** | **GET** /api/order/{OrderID}/questions | |
| **Headers:** | Content-Type | application/json |
| Authorization | Bearer: {token} |

|  |
| --- |
| **Example** |
| curl --request GET 'https://sandbox.parcelow.com/api/order/57/questions' \  --header 'Content-Type: application/json' \  --header 'Authorization: Bearer eyJ0eXAiOiJ.eyJnRpIjo.rLHS2' |

|  |
| --- |
| **Return** |
| {  "success": true,  "data": {  "questions":[  {  "id": 1,  "question": "Qual a sua data de nascimento?",  "answers": [  {"id": 1, "answer": "25/05/1990"},  {"id": 2, "answer": "05/11/1971"},  {"id": 3 "answer": "18/04/1982"}  ]  },  {  "id": 2,  "question": "Qual o nome da sua mãe?",  "answers": [  {"id": 4, "answer": "Maria"},  {"id": 5, "answer": "Luiza"},  {"id": 6, "answer": "Silvia"}  ]  }  ]  }  } |

**How to sent answers?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Request:** | **POST** /api/order/{OrderID}/questions/answers | | |
| **Headers:** | Content-Type | | application/json |
| Authorization | | Bearer: {token} |
| **Properties** | **Type** | **Mandatory** | **Description** |
| questions | key | yes | Questions Array |
| questions[0].id | integer | yes | Question ID |
| questions[0].answer | integer | yes | Answer ID |
| questions[1].id | integer | yes | Question |
| questions[1].answer | integer | yes | Answer ID |

|  |
| --- |
| **Example** |
| curl --request POST 'https://sandbox.parcelow.com/api/order/57/questions/answers' \  --header 'Content-Type: application/json' \  --header 'Authorization: Bearer eyJ0eXAiOiJ.eyJnRpIjo.rLHS2' \  --data-raw '{  "questions": [  {"id": 1, "answer": 1},  {"id": 2, "answer": 3}  ]  }' |

|  |
| --- |
| **Return** |
| {  "success": true,  "message": "Confirmado. Você já pode processar o pagamento."  } |

|  |
| --- |
| **Return Error: HTTP 400 status** |
| {  "success": false,  "message": "Identidade não confirmada."  } |

# Credit Card Payments

In order to process credit card payments in Transparent mode, it is necessary to confirm the customers' personal information from the previous step.

|  |  |  |  |
| --- | --- | --- | --- |
| **Request:** | **POST** /api/order/{OrderID}/payment | | |
| **Headers:** | Content-Type | | application/json |
| Authorization | | Bearer: {token} |
| **Properties** | **Type** | **Mandatory** | **Description** |
| d | text | yes | Fixed: "credit-card" |
| installment | text | yes | Number of payments, min:1 and max:12 |
| card | key | yes | data to send credit card information |
| card.number | integer | yes | credit card number |
| card.holder | text | yes | card holder name |
| card.exp\_month | integer | yes | credit card expiration month |
| card.brand | text | yes | type of credit card |
| card.exp\_year | integer | yes | credit card expiration year |
| card.cvv | integer | yes | credit card security numbers |
| card.address\_cep | Integer | yes | Billing Address Zip Code |
| card.address\_street | text | yes | Street of billing address |
| card.address\_number | integer | yes | Billing address number |
| card.address\_complement | text | no | billing address supplement |
| card.address\_neighborhood | text | yes | Neighborhood of billing address |
| card.address\_city | text | yes | City of billing address |
| card.address\_state | text | yes | Billing address status |

|  |
| --- |
| **Example** |
| curl --request POST 'https://sandbox.parcelow.com/api/order/57/payment \  --header 'Content-Type: application/json' \  --header 'Authorization: Bearer eyJ0eXAiOiJ.eyJnRpIjo.rLHS2' \  --data-raw '{  "method": "credit-card",  "installment": "1",  "card": {  "number": 1234123412341234,  "holder": "JOHN DOE",  "exp\_month": 1,  "exp\_year": 2021,  "cvv": 123,  "brand": "visa",  "address\_cep": "15478954",  "address\_street": "St. Test",  "address\_number": 45,  "address\_complement" : "Próximo do Supermercado",  "address\_neighborhood" : "Bairro",  "address\_city": "Cidade",  "address\_state": "SP"  }  }' |

|  |
| --- |
| **Return** |
| {  "data": true,  "message": "Pagamento realizado com sucesso.",  } |

|  |
| --- |
| **Return Error: HTTP 400 status** |
| {  "success": false,  "message": "Seu pedido foi submetido à análise de risco e não foi aprovada pelos analistas."  } |

|  |
| --- |
| **Return Error: HTTP 400 status** |
| {  "success": false,  "message": "Cartão não autorizado."  } |

# Check Transactions

Endpoint used to check updated data of a given transaction.

You can still obtain the updated URL for the Payment link in the event that the transactions are not finalized (canceled or non-authorized payments).

|  |  |  |
| --- | --- | --- |
| **Request:** | **GET** /api/order/{OrderID} | |
| **Headers:** | Content-Type | application/json |
| Authorization | Bearer: {token} |

|  |
| --- |
| **Example** |
| curl --request GET 'https://sandbox.parcelow.com/api/order/123' \  --header 'Content-Type: application/json' \  --header 'Authorization: Bearer eyJ0eXAiOiJ.eyJnRpIjo.rLHS2' |

|  |
| --- |
| **Return** |
| {  "data": {  "id": 62,  "reference": "my-order-ref-0001",  "disable\_email\_notifications": false,  "order\_amount": 250800,  "total\_usd": 279810,  "total\_brl": 1192874,  "installments": 1,  "order\_date": "2020-02-28T19:42:22.000000Z",  "status": 0,  "status\_text": "Open",  "client": {  "name": "John Doe",  "email": "johndoe@example.com",  "phone": "15987452584",  "cpf": "43164853858",  "cep": "18053190",  "created\_at": "2020-02-28T15:41:27.000000Z",  "updated\_at": "2020-02-28T19:11:31.000000Z",  "rg": "187236981726312",  "birthday": "1982-01-14T00:00:00.000000Z",  "address\_street": "Rua Raimundo Ribeiro dos Santos Filho",  "address\_number": "12",  "address\_complement": null,  "address\_neighborhood": "Conjunto Hab. Júlio de Mesquita Filho",  "address\_city": "Sorocaba",  "address\_state": "SP",  "is\_diferent\_card\_address": 0,  "card\_address\_cep": null,  "card\_address\_street": null,  "card\_address\_number": null,  "card\_address\_complement": null,  "card\_address\_neighborhood": null,  "card\_address\_city": null,  "card\_address\_state": null  },  "history\_log": [  {  "message": "Order created",  "created\_at": "2020-02-28T19:42:22.000000Z",  "updated\_at": "2020-02-28T19:42:22.000000Z"  }  ],  "items": [  {  "reference": "my-item-ref-0001",  "description": "Apple iMac 21.5-inch",  "quantity": 1,  "amount": 149900,  "created\_at": "2020-02-28T19:42:22.000000Z",  "updated\_at": "2020-02-28T19:42:22.000000Z"  },  {  "reference": "my-item-ref-0002",  "description": "Apple iPhone 11 Pro",  "quantity": 1,  "amount": 99900,  "created\_at": "2020-02-28T19:42:22.000000Z",  "updated\_at": "2020-02-28T19:42:22.000000Z"  },  {  "reference": "shipping",  "description": "Shipping",  "quantity": 1,  "amount": 1000,  "created\_at": "2020-02-28T19:42:22.000000Z",  "updated\_at": "2020-02-28T19:42:22.000000Z"  }  ],  "redirect\_success": null,  "redirect\_failed": null  "coupon": {  "code": "10DESCONTO",  "value": "10.00",  "issuer": "merchant\_api"  },  "url\_checkout": "https://sandbox.parcelow.com/payment/4openRe7Az/K4oeERva0B?expires=1582920747&signature=f1fb1d3337b95275d4e9adbc8d97cf0ce30848cda037c2ecf3c450d733131f59"  }  } |

|  |
| --- |
| **Return Error: HTTP 400 status** |
| {  "success": false,  "message": "Invalid Order ID"  } |

# Simulate Payment Options

Endpoint used to check the payment options and amounts of each payment.

|  |  |  |  |
| --- | --- | --- | --- |
| **Request:** | **GET** /api/simulate | | |
| **Headers:** | Content-Type | | application/json |
| Authorization | | Bearer: {token} |
| **Properties** | **Type** | **Mandatory** | **Description** |
| amount | decimal | yes | Amount: Total amount of the order to check payment options. |

|  |
| --- |
| **Example** |
| curl --request GET 'https://sandbox.parcelow.com/api/simulate?amount=1000' \  --header 'Content-Type: application/json' \  --header 'Authorization: Bearer eyJ0eXAiOiJ.eyJnRpIjo.rLHS2' |

|  |
| --- |
| **Return** |
| {  "data": {  "order": "1093.50",  "ted": {  "amount": "5900.30"  },  "creditcard": {  "installments": [  {  "installment": 1,  "monthly": "5900.30",  "total": "5900.30"  },  {  "installment": 2,  "monthly": "3030.26",  "total": "6060.52"  },  {  "installment": 3,  "monthly": "2038.19",  "total": "6114.56"  },  {  "installment": 4,  "monthly": "1542.23",  "total": "6168.92"  },  {  "installment": 5,  "monthly": "1244.72",  "total": "6223.60"  },  {  "installment": 6,  "monthly": "1046.43",  "total": "6278.59"  },  {  "installment": 7,  "monthly": "904.84",  "total": "6333.90"  },  {  "installment": 8,  "monthly": "798.69",  "total": "6389.52"  },  {  "installment": 9,  "monthly": "716.16",  "total": "6445.47"  },  {  "installment": 10,  "monthly": "650.17",  "total": "6501.73"  },  {  "installment": 11,  "monthly": "596.21",  "total": "6558.31"  },  {  "installment": 12,  "monthly": "551.27",  "total": "6615.21"  }  ]  }  }  } |

|  |
| --- |
| **Return Error: HTTP 400 status** |
| {  "success": false,  "message": "Invalid amount"  } |

# Webhooks

The system will send HTTP POST notifications to the registered URL as a webhook in the Client Access Registration phase

The notifications will only be sent if the transactions are created through the API

The notifications will be sent though the event:

event\_order\_paid  
event\_order\_declined  
event\_order\_confirmed  
event\_order\_canceled.

When the order status is modified, you will receive a JSON in the body of the request like this:

|  |
| --- |
| {  "order": {  "id": 62,  "reference": "my-order-ref-0001",  "disable\_email\_notifications": true,  "order\_amount": 250800,  "total\_usd": 279810,  "total\_brl": 1192874,  "installments": 1,  "order\_date": "2020-02-28T19:58:29.000000Z",  "status": 2,  "status\_text": "Paid",  "client": {  "name": "John Doe",  "email": "johndoe@example.com",  "phone": "15987452584",  "cpf": "43164853858",  "cep": "18053190",  "created\_at": "2020-02-28T15:41:27.000000Z",  "updated\_at": "2020-02-28T19:11:31.000000Z",  "rg": "187236981726312",  "birthday": "1982-01-14T00:00:00.000000Z",  "address\_street": "Rua Raimundo Ribeiro dos Santos Filho",  "address\_number": "12",  "address\_complement": null,  "address\_neighborhood": "Conjunto Hab. J. de Mesquita Filho",  "address\_city": "Sorocaba",  "address\_state": "SP",  "card\_address\_cep": null,  "card\_address\_street": null,  "card\_address\_number": null,  "card\_address\_complement": null,  "card\_address\_neighborhood": null,  "card\_address\_city": null,  "card\_address\_state": null  },  "history\_log": [{  "message": "Order created",  "created\_at": "2020-02-28T19:42:22.000000Z",  "updated\_at": "2020-02-28T19:42:22.000000Z"  }, {  "message": "Order paid by admin",  "created\_at": "2020-02-28T19:58:29.000000Z",  "updated\_at": "2020-02-28T19:58:29.000000Z"  }],  "items": [{  "reference": "my-item-ref-0001",  "description": "Apple iMac 21.5-inch",  "quantity": 1,  "amount": 149900,  "created\_at": "2020-02-28T19:42:22.000000Z",  "updated\_at": "2020-02-28T19:42:22.000000Z"  }, {  "reference": "my-item-ref-0002",  "description": "Apple iPhone 11 Pro",  "quantity": 1,  "amount": 99900,  "created\_at": "2020-02-28T19:42:22.000000Z",  "updated\_at": "2020-02-28T19:42:22.000000Z"  }, {  "reference": "shipping",  "description": "Shipping",  "quantity": 1,  "amount": 1000,  "created\_at": "2020-02-28T19:42:22.000000Z",  "updated\_at": "2020-02-28T19:42:22.000000Z"  }]  },  "redirect\_success": null,  "redirect\_failed": null,  "coupon": {  "code": "10DESCONTO",  "value": 1000,  "issuer": "merchant\_api"  },  "event": "event\_order\_paid",  "timestamp": "2020-02-28T19:58:33.488253Z"  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Possíveis Status** | | | |
| **ID do Status** | **Tipo de Status** | **Descrição** |
| **Status 0** | Open | Is the initial status of when an order is created. |
| **Status 1** | Confirmed | When the user accesses the payment screen, and informs their personal data, the order is CONFIRMED. |
| **Status 2** | Paid | When the user makes the payment, and it is captured by Cielo.  Or when the ADMIN, asynchronously, marks the order as paid. |
| **Status 3** | Canceled | When the order is cancelled by Parcelow, or Partner. |
| **Status 4** | Declined | When the order is marked as DECLINED by Parcelow.  Or when Cielo returns non-reversible code (no possibility of retry). |
| **Status 5** | Waiting Receipt | When the order is made by TED or PIX, it stays in this status until Parcelow analyzes and times as paid. |
| **Status 6** | Waiting Docs | When the order is worth more than USD $3,000.00, after payment confirmation, it enters this status until Parcelow confirms personal documents sent pelo comprador. |
| **Status 7** | In Review | When the order is suspected of credit card fraud, due to the cardholder's name being different from the buyer's name, until Parcelow decides to mark it as PAID or DECLINED. |
| **Status 8** | In AntiFraund Review | In AntiFraund Review  When Konduto (anti-fraud) marks the order as REVIEW, until Parcelow decides to mark it as PAID or DECLINED. |
| **Status 9** | Waiting Payment | Waiting Payment When Cielo does not authorize payment, it stays in this status Awaiting Payment, until the buyer makes a new payment attempt. |