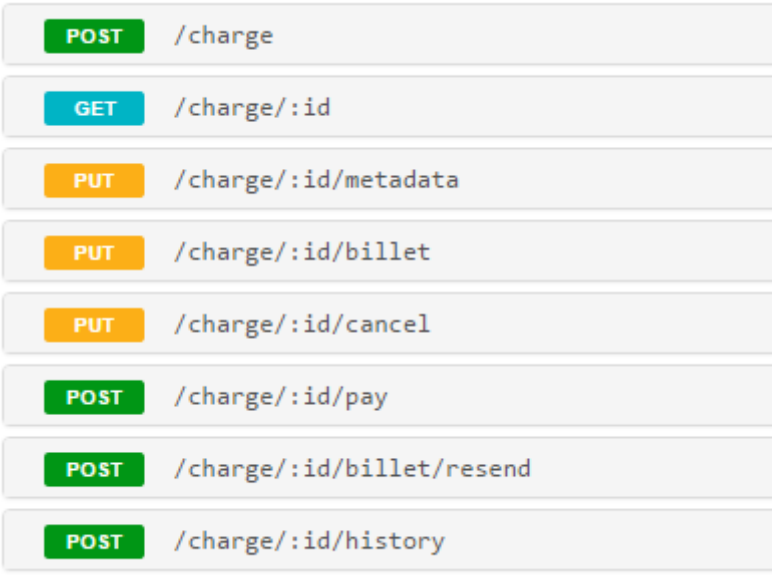
# Playground: Transações

A integração com a Gerencianet tem duas etapas: a primeira cria uma transação e a segunda define a forma de pagamento e os dados do comprador. Com a cobrança criada, já é possível definir a forma de pagamento como boleto ou cartão.

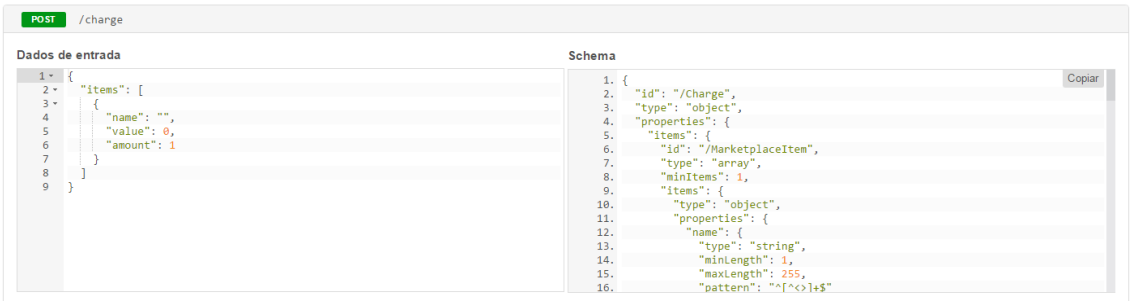
Em outras palavras, a geração da transação é o passo inicial, somente após a geração da transação (isto é, depois de gerada a cobrança) é que será definida a forma de pagamento (boleto bancário ou cartão de crédito).

Conforme pode-se observar abaixo, todos esses *endpoints* estão disponíveis na modalidade *"Transações"*:

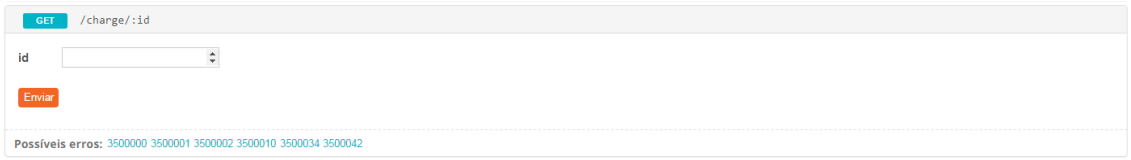



A seguir, vamos entender melhor cada um dos métodos disponíveis no ***"Playground"***:

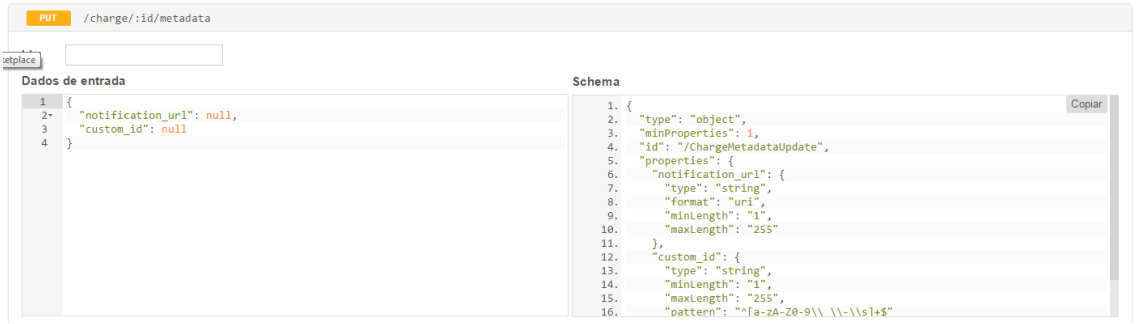
**POST** **/charge -** Permite criar uma nova transação; retorna um código identificador da transação denominado *“charge_id”.* Em outras palavras, somente após a geração da transação (isto é, depois de gerada a cobrança) é que vamos associar esta transação criada a um método de pagamento (boleto, cartão, etc).




**GET** **/charge/:id** - Busca via GET um *“charge_id”* de alguma transação e retorna dados referentes àquela transação. Por exemplo, após criar a transação (POST /charge), você pode inserir o *“charge_id”* correspondente para que seja retornada informações a respeito dessa transação foi criada.

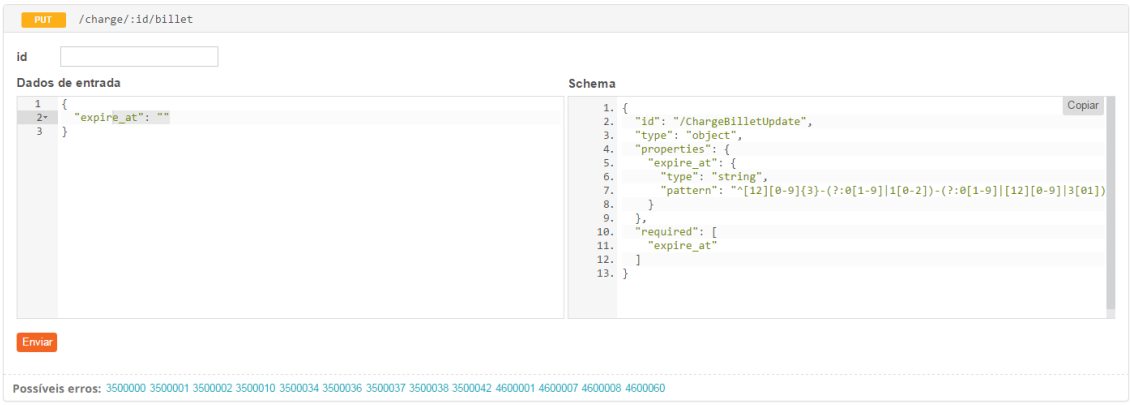



**PUT** **/charge/:id/metadata** - Diz ao servidor para enviar todas as requisições PUT para as URLs. O destino para todas as requisições PUT para esse script deve ser o próprio script, não o nome do arquivo enviado. Observe que você deve também enviar o código identificador da transação “charge_id”. O atributo “custom_id” possibilita você atribuir o seu código identificador (do seu próprio sistema) caso o tenha, à *“charge_id”* da API Gerencianet.

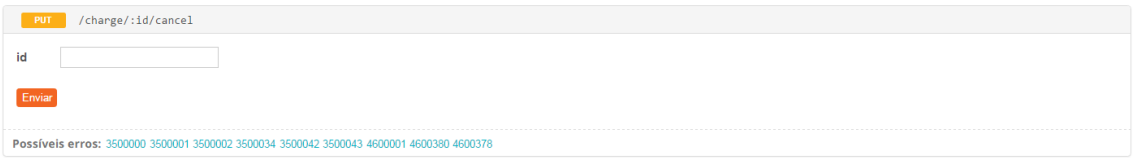



**PUT** **/charge/:id/billet** - Uma transação que possui "Boleto Bancário" como forma de pagamento definida e ainda não foi paga pode ter a data de vencimento alterada. O

formato da data de vencimento deve seguir o seguinte padrão: YYYY-MM-DD. Lembrando que é preciso informar a ID (“charge_id”) da transação correspondente.



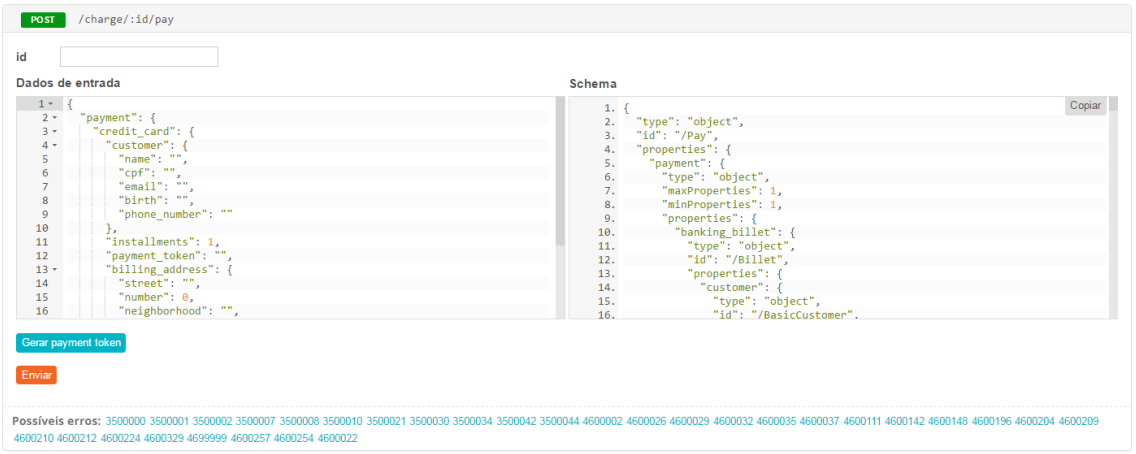

**PUT** **/charge/:id/cancel** - Através deste *endpoint* é possível cancelar uma transação criada. Para tal, é preciso informar a ID correspondente (“charge_id”) da transação. Somente transações com os status “new” (novo) ou “waiting” (aguardando) podem ser canceladas. A partir do momento que uma transação é cancelada, existe apenas uma condição para que esse status seja alterado novamente: se o cliente imprimir o boleto antes do integrador cancelar a transação, ele poderá realizar o pagamento normalmente em uma agência bancária. Nesse caso, o integrador e o pagador recebem a confirmação do pagamento como já acontece normalmente e o status da cobrança é alterado de *“canceled”* (cancelado) para *“paid”* (pago).



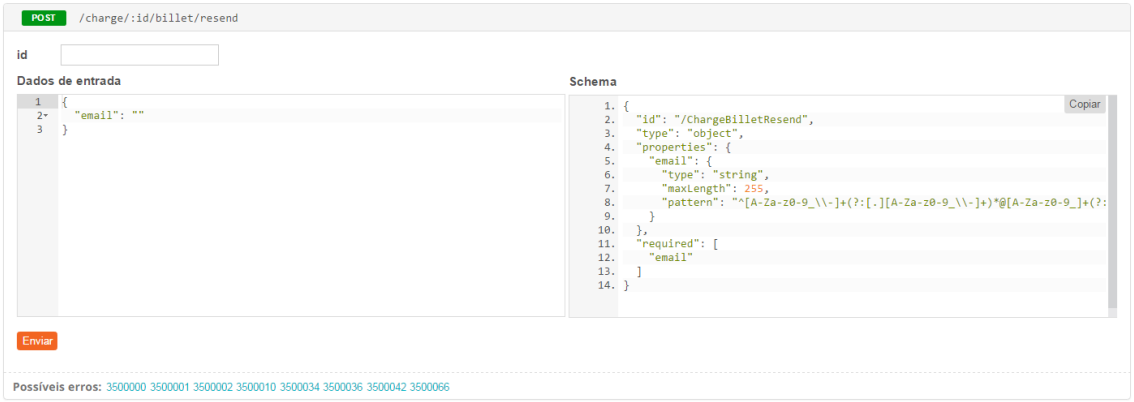

**POST** **/charge/:id/pay -** Após gerar uma transação (POST /charge), a mesma fica classificada com o status de “new” (novo), ou seja, uma nova transação foi gerada, porém, nenhum método de pagamento foi atribuído a ela. Para definir uma forma de pagamento para a transação criada, o integrador pode escolher entre “banking_billet” ou “credit_card” (boleto bancário e cartão de crédito, respectivamente).

1) *Via boleto bancário*: através do quadro “Schema” é possível observar quais informações são obrigatórias ou não. A transação passa por um ciclo de alteração de status, sendo criada inicialmente com o status de “new” (novo) e, ao definir a forma de pagamento (no nosso caso, boleto), o status passará a ser “waiting” (aguardando). Isso significa que o boleto foi gerado com sucesso, mas ainda não foi pago. Ao escolher boleto, a resposta do consumo já terá a linha digitável e um link para acessar o boleto. Lembrando que em nosso ambiente de testes não é possível simular o pagamento de um boleto.

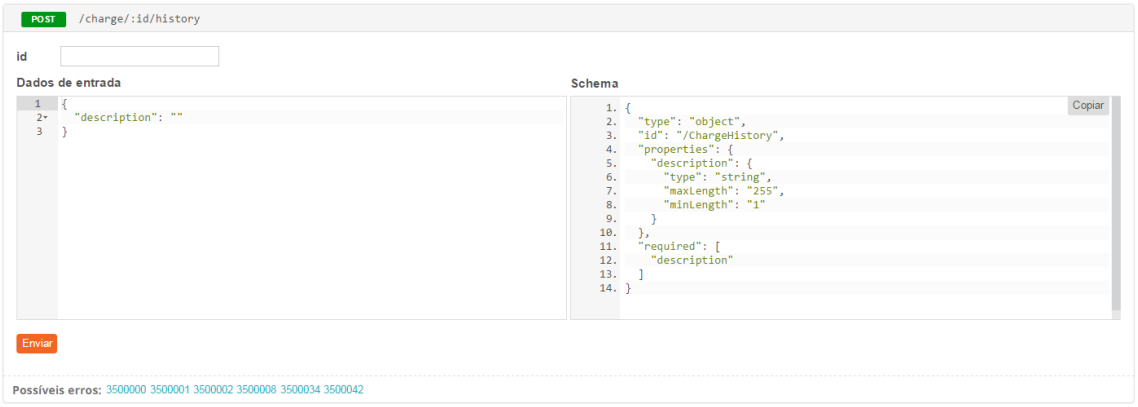
2) *Via cartão de crédito*: a principal diferença para o boleto está relacionada à necessidade de utilização de um código denominado *“payment_token”*. No ambiente de testes (“playground”), você pode gerar o seu token clicando no botão *“Gerar payment token”* e colá-lo dentro da tag “payment_token”. Já no ambiente de produção a obtenção do *payment_token* se dá pelo consumo de um código Javascript (veja mais neste link).




**POST** **/charge/:id/billet/resent -** permite o reenvio do boleto bancário para o e-mail desejado. É necessário que você informe a ID (“charge_id”) do boleto em questão.



**POST /charge/:id/history -** Acrescenta uma descrição a uma determinado transação. A chave ID identifica a transação desejada. Essa descrição deve possuir no mínimo um caractere e no máximo 255 caracteres.




# Playground: Carnês

Um carnê é um conjunto de transações (parcelas) geradas em lote e com forma de pagamento já definida. As parcelas de um carnê vencem mensalmente.

Para gerar um carnê, você necessita informar os seguintes dados: os itens (ou serviço), a data de vencimento da primeira parcela e o número de parcelas (repetições).

Se você quiser, também é possível definir algumas informações adicionais, como:

- Informações de metadata, como notification_url e/ou custom_id;
- Se o valor total deve ser dividido entre todas as parcelas ou se cada parcela deve possuir o valor;
- Inserir instruções no carnê (no máximo 4 linhas).

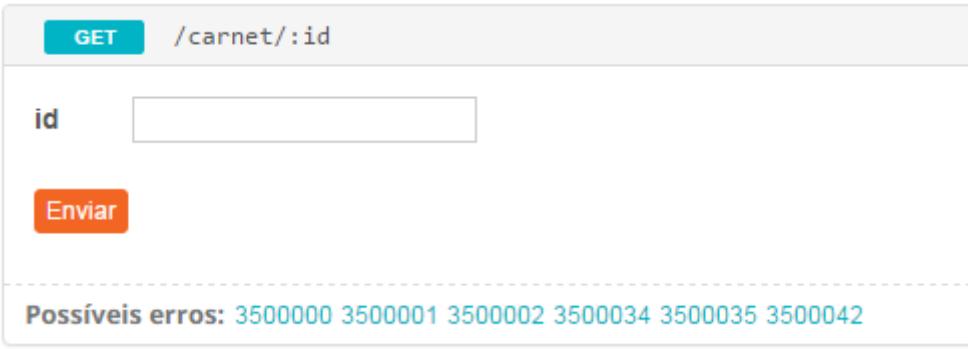
Todos os *endpoints* listados a seguir estão disponíveis na modalidade “Carnês” e podem ser consumidos:

| | |
|---|---|
| POST | /carnet |
| GET | /carnet/:id |
| PUT | /carnet/:id/metadata |
| PUT | /carnet/:id/parcel/:parcel |
| POST | /carnet/:id/resend |
| POST | /carnet/:id/parcel/:parcel/resend |
| POST | /carnet/:id/history |

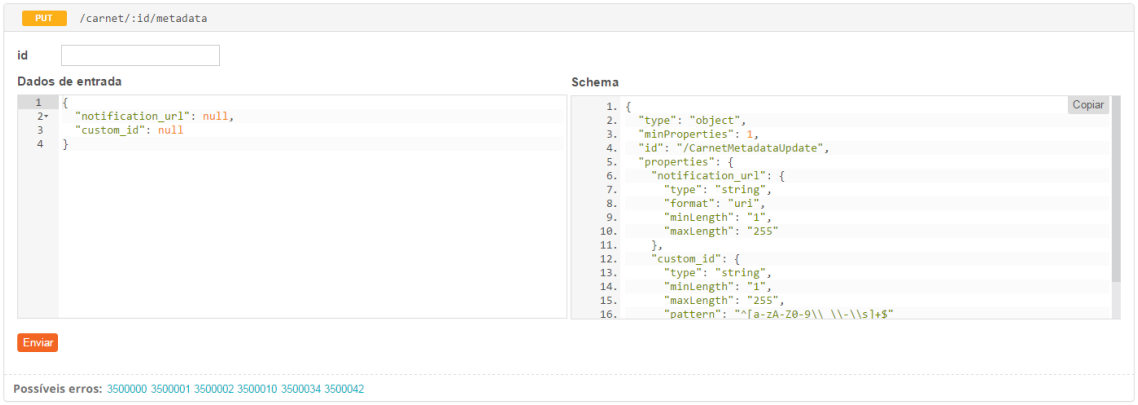
A seguir, vamos entender melhor cada um dos métodos disponíveis no ***"Playground"***:

**POST /carnet -** Este método é utilizado para geração de um conjunto de transações (parcelas) geradas em lote e com uma forma de pagamento já definida. As parcelas de um carnê vencem mensalmente, de acordo com o dia que você definir. Um carnê tem o status de "active" (ativo) até o vencimento da última parcela. Os principais atributos da geração de um carnê são: "item" (item que está sendo vendido), "customer" (dados pessoais do pagador), "expire_at" (vencimento da primeira parcela do carnê) e "repeats" (número de parcelas do carnê). Ainda cabe frisar que cada carnê gerado possui uma identificação única ("carnet_id"), e cada parcela desse carnê também possui um identificador único ("charge_id").

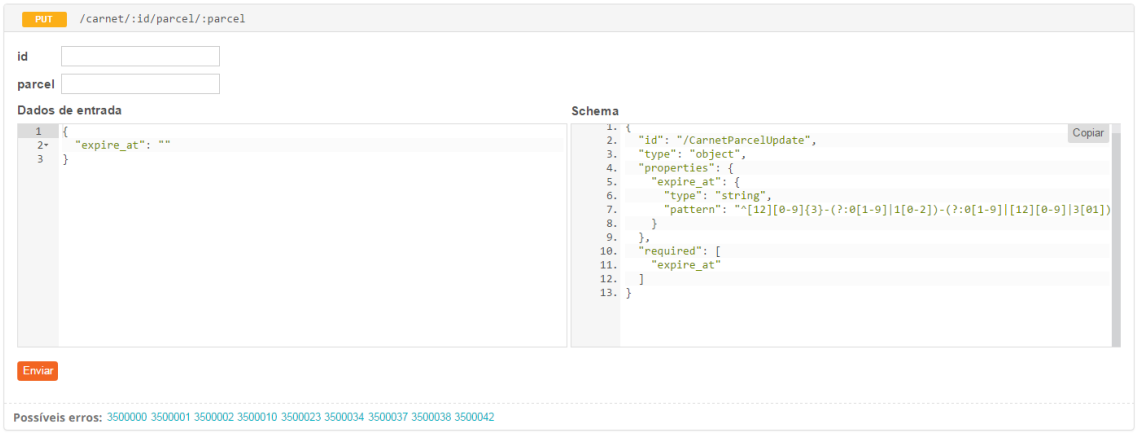

**GET /carnet/:id** - Retorna informações sobre um carnê criado. A busca é feita a partir do código identificador do carnê ("charge_id") do carnê desejado. Cada transação criada via carnê possui uma única chave identificadora que a identifica.



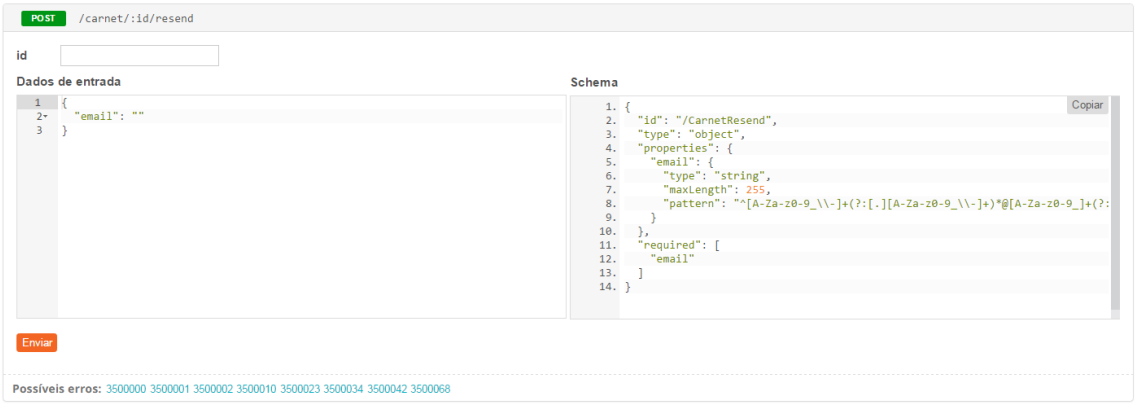

**PUT** **/carnet/:id/metadata** - Permite incluir em um carnê informações como “notification_url” e “custom_id”. Através da “notification_url” as confirmações, falhas ou qualquer tipo de alteração de status serão informadas à URL fornecida na geração da transação, e o “custom_id” possibilita que você associe seu código identificador (do seu próprio sistema ou aplicação) ao “carnet_id” da API Gerencianet. Lembrando que “custom_id” é de cunho opcional.



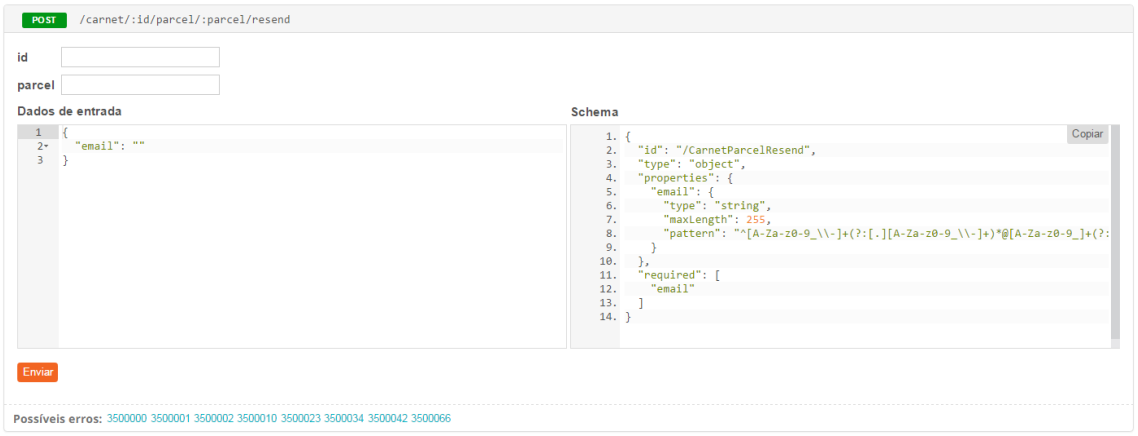

**PUT** **/carnet/:id/parcel/:parcel** - Possibilita alterar o vencimento de uma determinada parcela de um determinado carnê e diz ao servidor para atualizar a data de vencimento. Para tal, é necessário que você informe o ID do carnê (“carnet_id”), qual a parcela que você atualizar o vencimento e a nova data de vencimento da parcela (“expire_at”). Somente parcelas que estejam com status “waiting” (aguardando) ou “unpaid” (não pago) podem ter suas datas de vencimento alteradas.



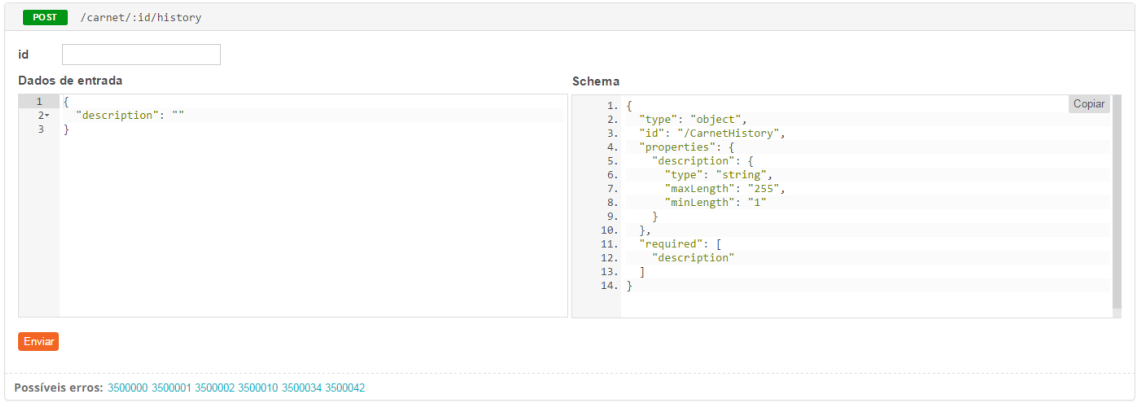

**POST /carnet/:id/resend -** Este método possibilita que o carnê seja reenviado para um endereço de email. É preciso que você informe o ID (“carnet_id”) do carnê desejado. Lembrando que neste *endpoint* todo o carnê será reenviado por email.




**POST /carnet/:id/parcel/:parcel/resend -** Este método possibilita que uma determinada parcela, de um determinado carnê seja reenviado para um endereço de email. É preciso que você informe o ID (“carnet_id”) do carnê desejado e qual parcela deseja que seja reenviada. Lembrando que neste *endpoint* somente a parcela informada será reenviada por email.




**POST** **/carnet/:id/history -** Este método permite acrescentar uma descrição a uma determinado transação via carnê. A chave “ID” (“carnet_id”) identifica a transação desejada. Essa descrição deve possuir no mínimo um caractere e no máximo 255 caracteres.



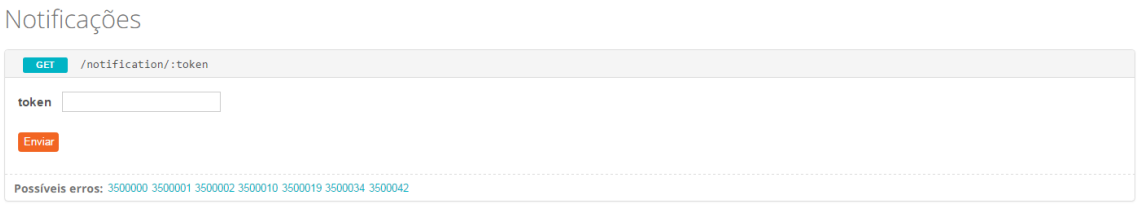

# Playground: Notificações

Uma transação gerada por meio da API pode passar por diversas alterações de status conforme interações do pagador, do integrador ou das operadoras e instituições bancárias envolvidas. Para acompanhar essas mudanças, é necessário preparar seu sistema para receber as notificações enviadas pela Gerencianet.

Para definir uma url que receberá as notificações de uma transação, é necessário cadastrá-la no momento da geração da cobrança ou posteriormente, utilizando o método de alteração de url de notificação.

O processo de notificação é realizado em duas etapas para garantir a segurança dos dados informados. Na primeira, seu sistema é avisado que houve uma alteração relacionada a uma transação. Na segunda, seu sistema consulta a Gerencianet para saber detalhes sobre essa alteração.

**GET** **/notification/:token** - Retorna o histórico de notificações enviados a uma determinada transação.




# Playground: Assinaturas

Uma assinatura é um conjunto de transações geradas de forma recorrente.

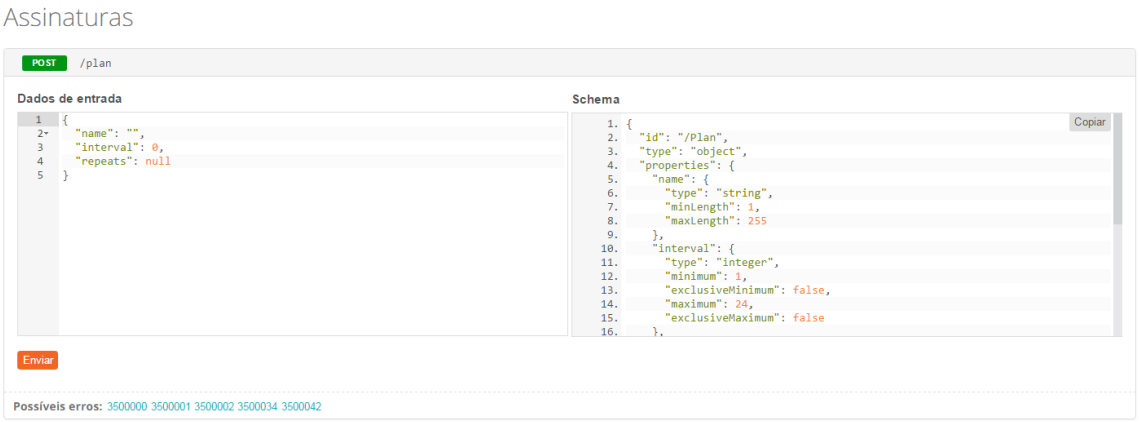
Para gerar uma assinatura, basta gerar uma cobrança e enviar, junto às informações dessa cobrança, dados que determinam o número de parcelas e a periodicidade em que o sistema deve gerar uma transação exatamente igual à primeira. Essas informações são denominadas Planos de Assinaturas.

Dessa forma, uma assinatura está sempre associada a um plano.

Em outras palavras, uma assinatura é sempre uma cobrança recorrente e pode ser cobrada por boleto bancário ou cartão de crédito. No caso de cartão, o cliente realiza um pagamento, e a cada “x” dias (definido pelo integrador) aquele mesmo valor é cobrado automaticamente do cartão de crédito. Já uma assinatura de boleto gera uma cobrança a cada “x” dias e envia para o email do cliente informado pelo integrador no momento da geração da assinatura.

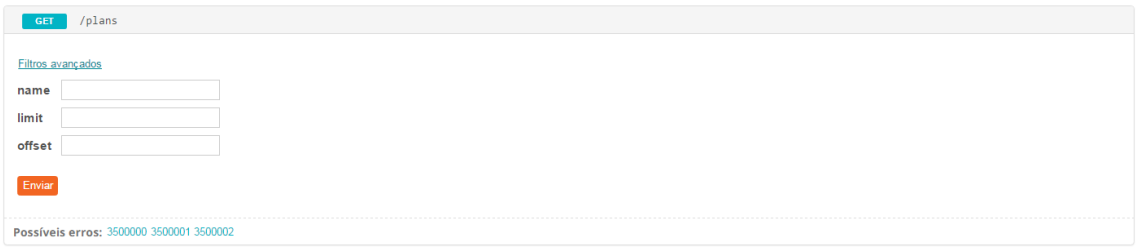
Todos os *endpoints* listados a seguir estão disponíveis na modalidade “Assinaturas” e podem ser consumidos:

**POST** **/plan** - Cria o plano de assinatura, em que o integrador define o nome do plano, intervalo (em meses) em que a assinatura será gerada e a quantidade de repetições que a cobrança será gerada.

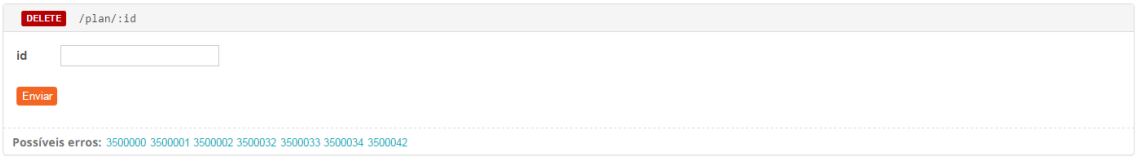



**GET** **/plans** - Busca informações relacionadas à uma assinatura. Existem filtros avançados que podem ser utilizados para localizar, tais como:

- *Name*: retorna resultados a partir da procura pelo nome do plano cadastrado previamente;
- *Limit*: limite máximo de registros de resposta;
- *Offset*: determina a partir de qual registro a busca será realizada.

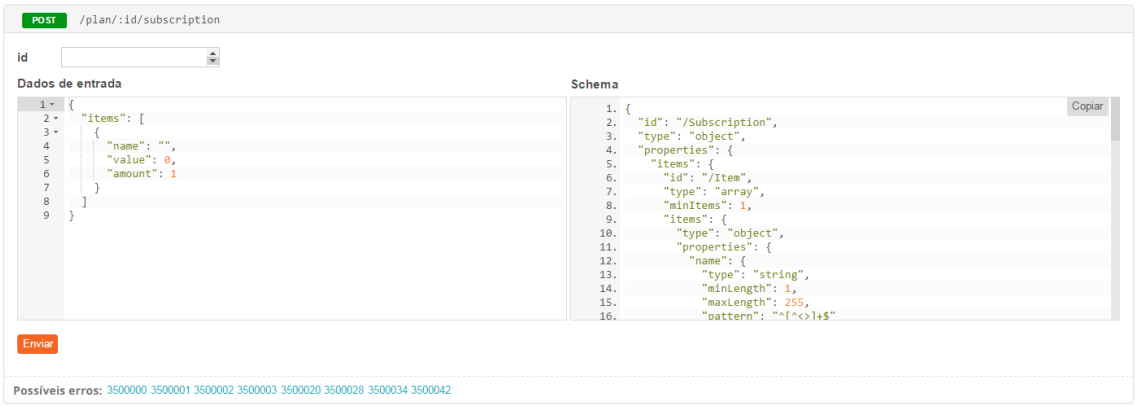



**DELETE** **/plan/:id** - Permite cancelar um plano de assinatura através de sua ID.

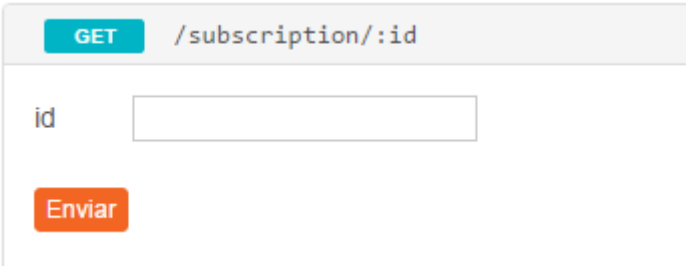



**POST** **/plan/:id/subscription** - Cria uma inscrição quando você precisa cobrar de forma recorrente seus clientes. Desta forma, os custos subsequentes serão criados

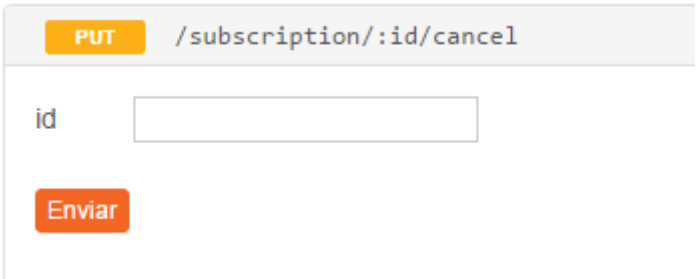
automaticamente com base na configuração do plano. Para tal, você deve informar a ID do plano criado previamente no qual deseja associar.



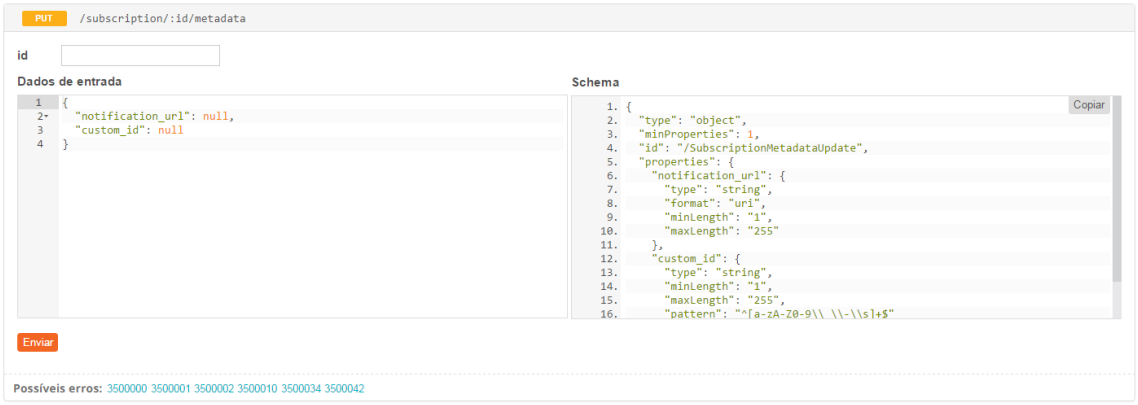

**GET** **/subscription/:id** - Possibilita buscar informações relacionadas à inscrição de uma assinatura vinculada a um plano.



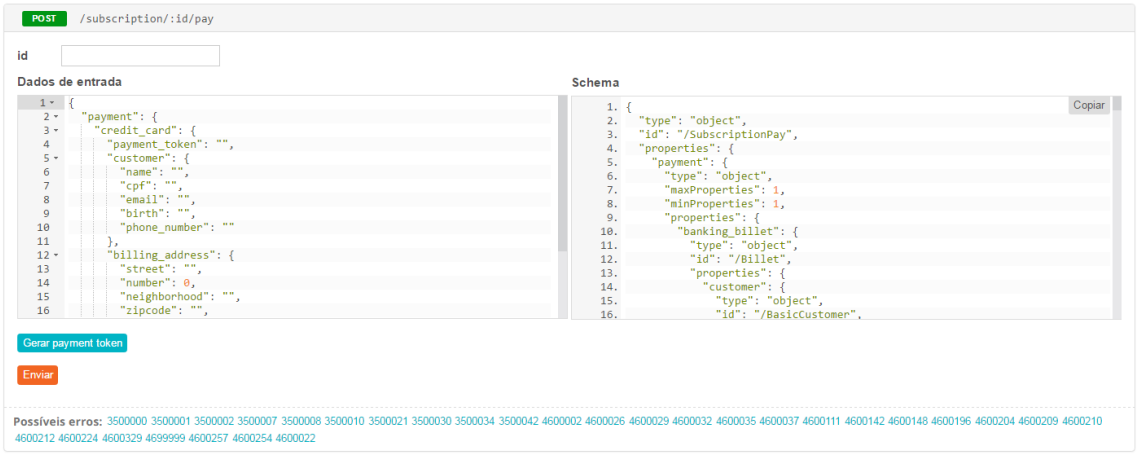

**PUT** **/subscription/:id/cancel** - Permite cancelar inscrições ativas em um plano de assinaturas. Para tal, deve-se usar como parâmetro o ID da inscrição que deseja cancelar.



**PUT** **/subscription/:id/metadata** - Atualiza o status de uma determinada inscrição e envia uma notificação para a URL cadastrada no sistema




**POST** **/subscription/:id/pay** - Define uma forma recorrente de pagamento à uma determinada assinatura. Ela poderá ser por cartão de crédito ou boleto bancário. No caso de cartão, fará débitos em seu cartão mensalmente de acordo conforme o número de repetições definido pelo plano, ou boleto que será gerado conforme o número de repetições definido pelo plano, podendo ser enviado por email. O assinante ou o vendedor podem cancelar a assinatura a qualquer momento. Quando isso ocorre, os dois são avisados via email, com todos os detalhes do cancelamento.

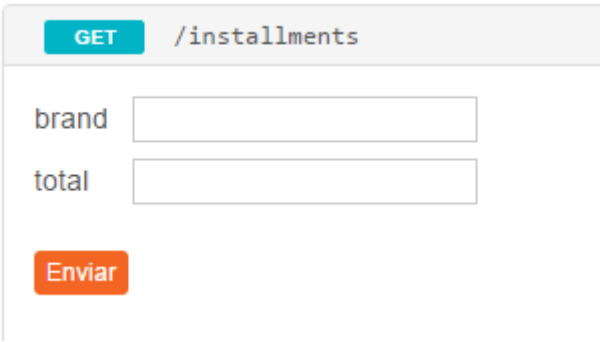


# Playground: Outros

O endpoint “installment” é utilizado para listar as parcelas de cada bandeira de cartão de crédito, já com os valores de juros e número de parcelas calculados de acordo com a conta integradora. Ou seja, se sua conta possui uma configuração de juros de cartão (opção disponível para clientes que optaram por receber valores de cartão de forma parcelada), não é necessário fazer nenhum cálculo, esse endpoint já informa os valores calculados.

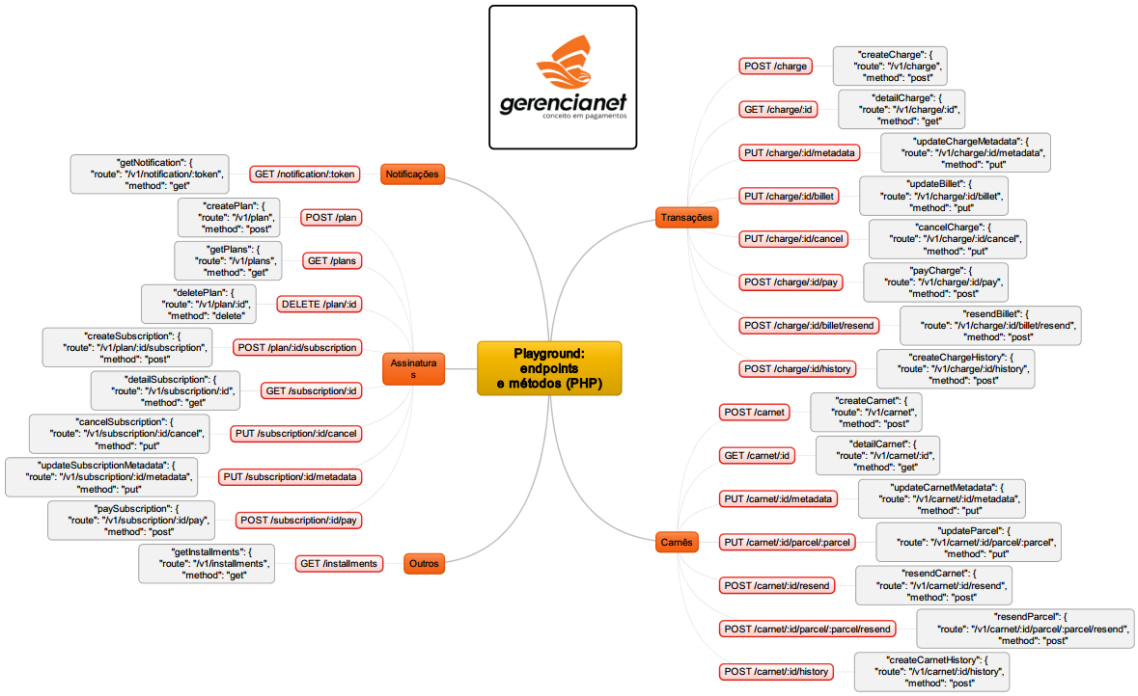
Na SDK de PHP por exemplo, a função que utiliza esse endpoint é chamada *getInstallments*, e um exemplo pode ser visualizada no GitHub.

**GET** **/installments** - Permite listar as parcelas de cada bandeira de cartão de crédito, já com os valores de juros e número de parcelas calculados de acordo com a conta integradora.



# Relacionamento entre endpoints do Playground e funções da SDK em PHP

A seguir é possível visualizar a relação entre os endpoints consumidos no Playground (“sandbox”) com o nome das funções utilizadas pela SDK em PHP. Dessa forma, após conhecer o mecanismo do playground, será possível associar os métodos e funções.




Para visualizar o arquivo completo, baixe-o neste link, em formato PDF.