Gift Cards

Cloud, Corporate*

The `giftcardTransAdd` mutation endpoint can be used to create gift card transactions. These transactions can be created immediately, or used with the `giftcardTransCapture` endpoint to support an authorization and capture approach that is similar to credit card payment gateways.

To use these endpoints, the caller must have permission to create transactions of the requested transaction type. This type of granularity can help restrict API clients to only creating transactions of certain types. The permissions are:

- `GiftCardTransAdd`
  - The caller can create a positive ADD transaction.
  - This transaction adds funds to a card, e.g. in response to a customer purchasing or topping up a card.
- `GiftCardTransAdjust`
  - The caller can create a positive or negative ADJUST transaction.
  - This transaction makes an adjustment entry to card balance, intended for internal systems.
- `GiftCardTransExternal`
  - The caller can create a positive or negative EXTERNAL transaction.
  - This transaction type is intended for external systems that create their own gift cards, such as ecommerce systems.
- `GiftCardTransImport`
  - The caller can create a positive or negative IMPORT transaction.
  - This transaction type is intended for importing balances from gift cards that have been transferred from other systems.
- `GiftCardTransUse`
  - The caller can create a negative USE transaction.
  - This transacdtion removes funds from a card, e.g. in response to a customer applying gift card funds towards a retail purchase.

[illegible]

[illegible]

This example creates a transaction that is recorded and completed immediately:

```
mutation AddTrans {
  giftcardTransAdd(
    trans: {
      codeType: PHYSICAL,
      code: "GC49288330",
      pin: "1234",
      transType: USE,
      amount: "-6.49",
      sourceId: "Api-test",
      refNo: "TRANS001"
  } ) {
    id
    balance
    giftcardId
  }
}
```

Note that:

- While the `codeType` parameter is optional, it is recommended especially if `code` values are not unique across all types of codes.
- Physical cards may have a PIN. The `pin` is only required if the transaction uses funds from the card (the transaction amount is negative).
- It’s recommended to set the `sourceId` and `refNo` parameters; these can help reconcile transactional information in the API client with POS gift card transactions.
  - The `refNo` should indicate the transaction number in the source system. These should be unique, although POS does not enforce uniqueness or otherwise validate the number.
  - The `sourceId` identifies the source system or agent that is creating the transaction – e.g. “Register 1”, “Ecommerce”. Reports can use the source ID to group transactions by system or agent.

[illegible]

This example authorizes a transaction that should be captured (or cancelled) in a subsequent call:

```
mutation AddTrans {
  giftcardTransAdd(
    trans: {
      codeType: PHYSICAL,
      code: "GC49288330",
      pin: "1234",
      transType: USE,
      amount: "-1.77",
      sourceId: "Api-test",
      refNo: "TRANS002",
      authOnly: true
  } ) {
    id
    authCode
    balance
    giftcardId
  }
}
```

Authorization is the same as an immediate transaction, except:

- The caller should save the returned `id` and `authCode` values; these will be needed in the subsequent call to capture or cancel the transaction.
- The gift card balance is updated only if the transaction uses funds (a negative transaction amount); this effectively reserves the funds.
- The gift card balance is not immediately updated if the transaction adds funds; this is only done after capture.

After a transaction authorization is created, the transaction must be captured:

```
mutation CaptureTrans {
  giftcardTransCapture(
    trans: {
      id: "80002",
      authCode: "88d2d781-18a6-4912-a543-7b2b18203db2"
  } ) {
    id
    balance
    giftcardId
  }
}
```

With the authorization process:

- The gift card balance will be updated if the transaction was authorized to add funds.
- The `sourceId` and `refNo` parameters can be supplied to overwrite any values that were originally specified with the original authorization; this may be helpful if these values are not known until capture.

Alternately, the transaction authorization can be cancelled:

```
mutation CancelTrans {
  giftcardTransCapture(
    trans: {
      id: "80003",
      authCode: "453eb370-528e-43be-b1c2-7aa9e70b254e",
      cancel: true
  } ) {
    id
    balance
    giftcardId
  }
}
```

Cancellation takes the same parameters as authorization, except for the inclusion of the `cancel` parameter.