Add Transaction Using Signed Requests

#server-to-server #pos

Use Cases

The following is best for integrating Spendgo into:

- POS systems
- E-commerce platforms
- Payments devices
- Customer-facing displays
- Self checkout kiosks

Endpoints

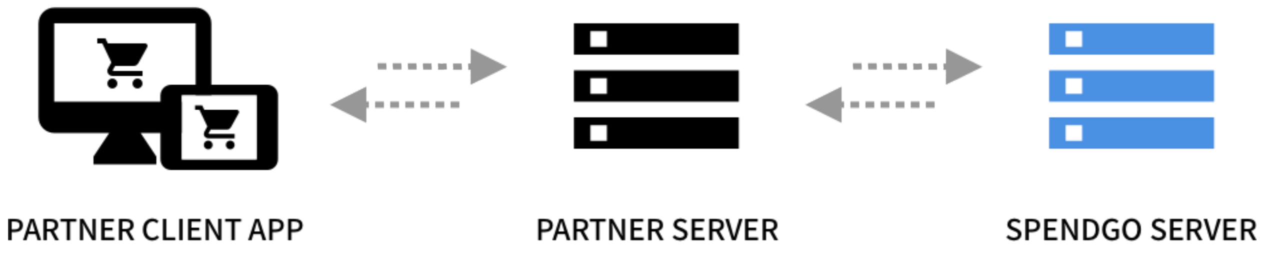
When enabling loyalty in a commerce system these are the API endpoints that are most commonly used:

POST /loyalty/accounts Creates a new member or retrieves an existing member.
GET /loyalty/accounts/{spendgo_id}/rewards?store={store_code} Retrieves member's available rewards for a specific store location.
GET /loyalty/accounts/{spendgo_id}/balance Retrieves member's points balance.
POST /loyalty/accounts/{spendgo_id}/orders Creates an order.

See API Reference document.

Security Requirements

These are deemed to be server-to-server integrations and the following diagram depicts the typical data flow between the Partner and Spendgo systems:



Getting Your Credentials

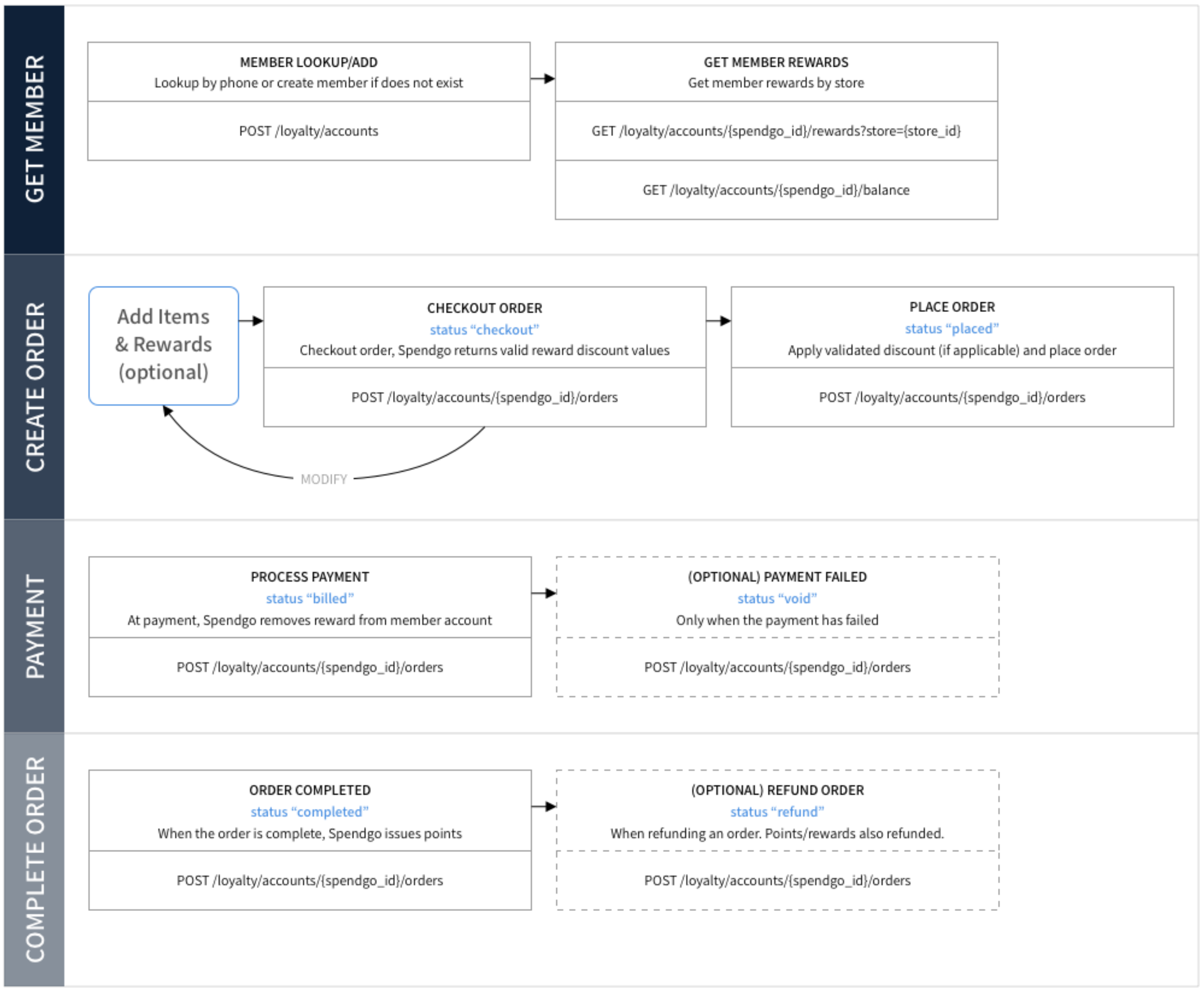
You use different types of security credentials depending on how you interact with Spendgo. For signed requests, Spendgo will share with you the following credentials:

- API Key
- Shared Client Secret
- Account ID

Note: *Security credentials are environment and account-specific. If you have access to multiple Spendgo accounts, use the credentials that are associated with the account and environment that you want to access.*

Customer Experience

A typical process workflow may look like the following:


GET MEMBER
MEMBER LOOKUP/ADD
Lookup by phone or create member if does not exist
POST /loyalty/accounts
GET MEMBER REWARDS
Get member rewards by store
GET /loyalty/accounts/{spendgo_id}/rewards?store={store_id}
GET /loyalty/accounts/{spendgo_id}/balance
CREATE ORDER
Add Items & Rewards (optional)
CHECKOUT ORDER
status "checkout"
Checkout order, Spendgo returns valid reward discount values
POST /loyalty/accounts/{spendgo_id}/orders
PLACE ORDER
status "placed"
Apply validated discount (if applicable) and place order
POST /loyalty/accounts/{spendgo_id}/orders
MODIFY
PAYMENT
PROCESS PAYMENT
status "billed"
At payment, Spendgo removes reward from member account
POST /loyalty/accounts/{spendgo_id}/orders
(OPTIONAL) PAYMENT FAILED
status "void"
Only when the payment has failed
POST /loyalty/accounts/{spendgo_id}/orders
COMPLETE ORDER
ORDER COMPLETED
status "completed"
When the order is complete, Spendgo issues points
POST /loyalty/accounts/{spendgo_id}/orders
(OPTIONAL) REFUND ORDER
status "refund"
When refunding an order. Points/rewards also refunded.
POST /loyalty/accounts/{spendgo_id}/orders

To integrate the above customer experience, use the following flow:
Note: for guest transactions, you can skip ahead to step 3.

GET MEMBER

1. Member lookup

How you capture the customer phone number can be determined by your application. Once you have the customer phone number, your server makes a Create/Retrieve a member call:

```
POST /loyalty/accounts
```

- If the member is found, a successful response will return the member's Spendgo unique id and account status.
- If the member did not already exist, the member is created. The member must complete the email verification process for their member account to be created.

2. Get member rewards

To get the member's available rewards that can be redeemed at a store, your server makes a Retrieve member rewards by store call:

```
GET /loyalty/accounts/{spendgo_id}/rewards?store={store_code}
```

- If the member has rewards available, you will get a response with the reward details. You will use these reward details later when redeeming a reward.

Optionally, you can also get the member's points balance by making a separate Retrieve member balance call.

```
GET
https://webservice.<ENVIRONMENT>.com/loyalty/accounts/{spendgo_id}/balance
```

CREATE ORDER

3. Add items & rewards

All items have been added to the cart and any rewards to be redeemed have been selected (only one reward can be redeemed per order).

4. Checkout order
At this stage your server makes an Orders call with a "checkout" status including the member's `spendgo_id` and the reward object (if a reward is being redeemed). The "checkout" order status is the first state in the ordering workflow.

```
POST
https://webservice.<ENVIRONMENT>.com/loyalty/accounts/{spendgo_id}/orders
```

Note: *If there has been no member sign-in up to this point, and you're creating a guest transaction, the `spendgo_id` parameter in the endpoint can be passed as "guest" and the order will not be linked to a member account.*

- Spendgo will validate the cart contents against the applied reward rules to confirm it meets the requirements. If the cart item(s) and reward are valid, the discount amount will be calculated and returned in the response. The validated discount* amount will be used in subsequent order calls — "placed", "billed", and "completed" — in this sequence.
  * When applying the discount value, it should match the value returned after the checkout call except with a Shop With Points reward. See Shop With Points for more details.

- If the cart item(s) and reward do not meet the requirements, a detailed error will be returned.

5. Place order
At this state the customer has submitted their order and your server makes the Orders call with a "placed" status. Always include all of the cart details and reward details (if applicable).

PAYMENT

6. Process payment
At this state the payment has been processed successfully. Your server makes the Orders call with a "billed" status and Spendgo removes any applied rewards from the member's account. In some use cases — such as advance ordering — the "billed" state may occur some time in the future, where payment is processed at a later time.

- A payment failure during this stage should result in the complete rollback of the transaction and triggering of the "voided" workflow. Spendgo will return the reward.

COMPLETE ORDER

7. Completed order

The “completed” status is the fourth and final state of the ordering workflow. This state is reached after the order has been confirmed, processed, and marked as shipped. When your server makes the Orders call with a “completed” status, Spendgo awards the points to the member’s account.

- If a refund is requested, your server makes the Orders call with a “refunded” status and any rewards redeemed will be returned to the account and points earned will be removed.