Add Transaction With OAuth 2.0

#server-to-server #oauth2.0

Use Cases

The following is best for integrating Spendgo into:

- E-commerce
- Order Ahead
- Mobile applications
- Pay-at-the-table solutions
- Reservation systems

Endpoints

When enabling loyalty in a commerce app which requires the user to sign in, these are the API endpoints that are most commonly used:

GET /oauth2/v1/token
Creates a member access token.

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.

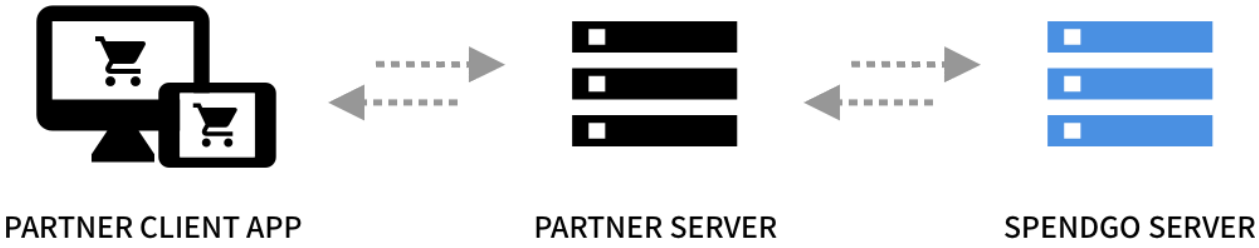
POST /loyalty/accounts/{spendgo_id}/orders
Creates an order.

See API Reference document.

Security Requirements

The Spendgo OAuth 2.0 is a client-side integration.

The loyalty APIs — Retrieve member rewards by store, Retrieve member balance, and Orders — 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 example, an OAuth 2.0 and loyalty integration will use separate credentials. Spendgo will share with you the following credentials for each interaction type:

- Client ID
- Shared Client Secret
- Account ID

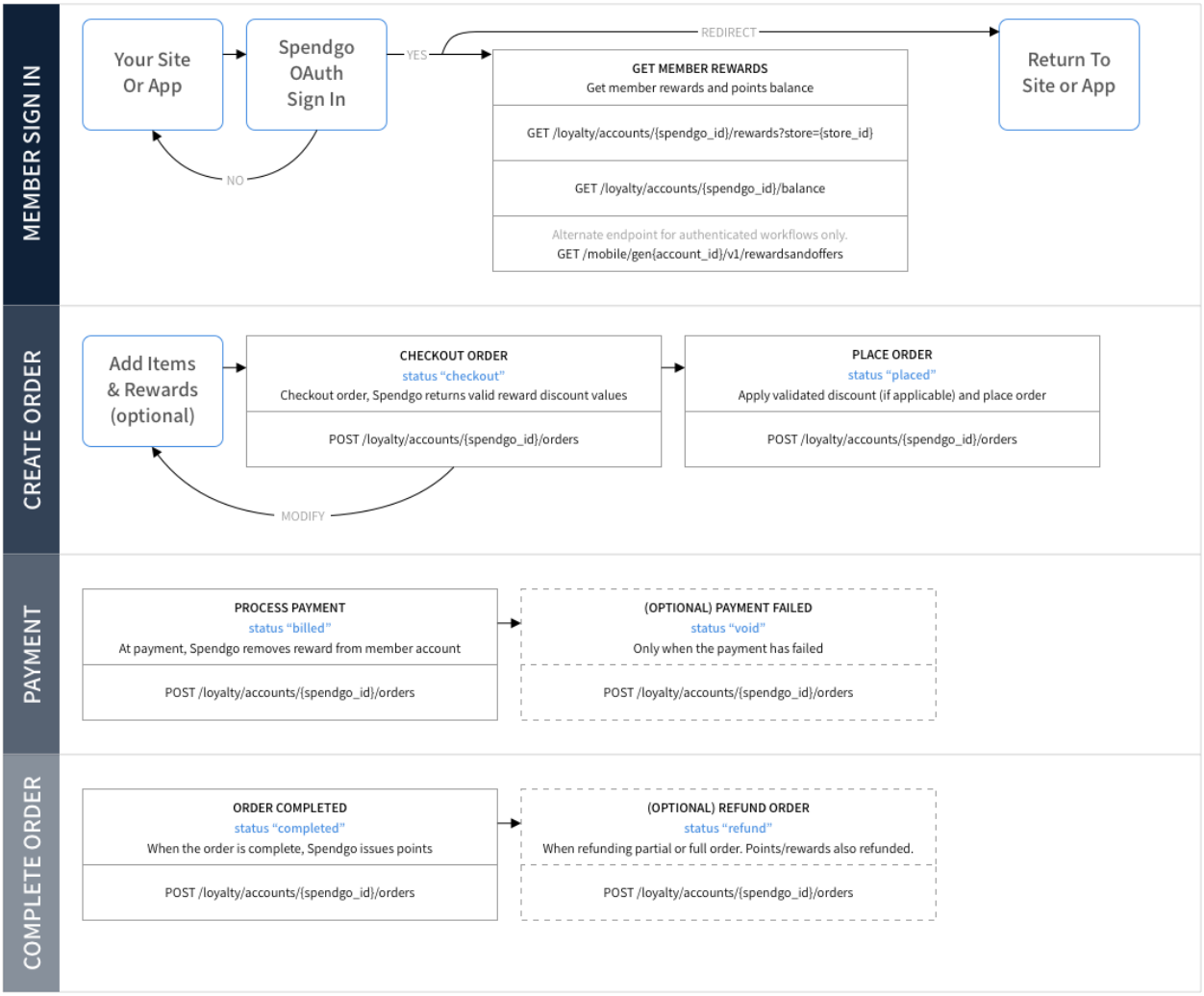
Note: *Security credentials are environment and account-specific. If you have access to multiple Spendgo accounts, use the credentials that are associated with the account and environment that you want to access.*

For OAuth 2.0, you with share with Spendgo:

- Whitelist URLs

Customer Experience

A typical customer experience may look like the following:



To integrate the above customer experience, use the following flow:
Note: for guest transactions, you can skip ahead to step 4.

1. User login

When a user initiates the log in, follow the Spendgo Identity & OAuth 2.0 flow to authenticate the user.

If the user does not sign in and is continuing as a guest, skip ahead to Step 4: Checkout order.

2. Get member rewards

To display the member's available rewards that can be redeemed at a store, your application server will make 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
```

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 application server will make 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 stage 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 Points Banking & Shop With Points rewards. See Points Banking & 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 you will make the Orders call with a "placed" status. Always include all of the cart details and reward details (if applicable).

6. Process payment

At this state the payment has been processed successfully. You will make 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.

7. Complete order

The "completed" status is the fourth and final stage of the ordering workflow. This state is reached after the order has been confirmed, processed, and marked as shipped. When you make the Orders call with a "completed" status, Spendgo awards the points to the member's account.

- If a refund is requested, you will make the Orders call with a "refunded" status and any rewards redeemed will be returned to the account and points earned will be removed.